

**UNIVERSIDAD AUTONOMA DE MADRID**

**ESCUELA POLITECNICA SUPERIOR**



**Grado en Ingeniería Informática**

## **TRABAJO FIN DE GRADO**

**PLATAFORMA WEB PARA LA GESTION DE UN GRUPO  
DE INVESTIGACIÓN**

**Miguel Macías Díaz  
Tutor: Simone Santini**

**Enero 2017**



# **PLATAFORMA WEB PARA LA GESTION DE UN GRUPO DE INVESTIGACIÓN**

**AUTOR: Miguel Macías Díaz**

**TUTOR: Simone Santini**

**Dpto. Ingeniería Informática  
Escuela Politécnica Superior  
Universidad Autónoma de Madrid  
Enero 2017**



# Resumen

Este Trabajo de Fin de Grado tiene como objetivo facilitar a los grupos de investigación la gestión de su trabajo (proyectos en los que participan los investigadores, colaboraciones externas, etc.) mediante el desarrollo de una aplicación web basado en el Framework FuelPHP.

La aplicación Web permite la gestión de los principales aspectos de un grupo de investigación como las personas que participan en el proyecto, las personas que colaboran con el proyecto mediante una colaboración externa, los proyectos que existen y los datos asociados a él, permitiendo que estos datos contengan formato multimedia con metadatos. También permite la gestión de informes y artículos donde se evita una edición en paralelo para evitar conflictos entre los participantes del proyecto. Por último, se ha añadido la capacidad de crear eventos para una mejor organización entre el grupo de trabajo del proyecto.

Para lograr una mejor gestión se han distinguido cuatro tipos de roles de usuario: *Administrador de la aplicación, Responsable del Proyecto, Miembro y Colaborador Externo*. Cada uno de estos roles podrá realizar distintas acciones en la aplicación basado en el nivel de importancia del Rol.

En esta aplicación se ha utilizado las últimas tecnologías respecto al diseño de aplicaciones Web, como el framework *FuelPHP*, *Bootstrap* y *Twig*. La aplicación Web ofrece una interfaz totalmente *responsive* con el objetivo de ser, a la vez, sencilla y profesional. Esta aplicación Web se basa en la arquitectura de software *MVC* (Modelo-Vista-Controlador) para conseguir una mejor reutilización del código y facilitar la tarea de mantenimiento en el futuro.

Se ha dividido la aplicación Web en tres subsistemas, llevando a cabo para cada uno de ellos, todas las fases del ciclo de vida de un software siguiendo un *modelo en cascada*.

Finalmente se han realizado pruebas para asegurar el cumplimiento de los requisitos funcionales y pruebas de carga y estrés para asegurar la robustez de la aplicación web desarrollada en este Trabajo de Fin de Grado.

## Palabras clave

Aplicación web, Bootstrap, Framework, FuelPHP, Gestión de un grupo de investigación, Metadatos, Modelo en cascada, Modelo-Vista-Controlador (MVC), Proyectos, Responsive, Twig.

# Abstract

This work aims at developing an application based on the FuelPHP Framework to help research groups in their work (among others: projects in which researchers are involved, external collaborations).

This web application allows the management of the main aspects of a research group, such as the people involved in a project, people who collaborate on a project through an external collaboration, projects that exist in the application and data associated with them. Likewise, the application also allows the management of reports and articles with facilities to evade conflicts between the people who try to edit a document at the same time. Finally, the application allows the creation of events for a better organization among researchers.

In order to achieve a better management, four types of user roles have been distinguished: *Application Manager*, *Project Manager*, *Member* and *External Collaborator*. Each of these roles can perform different actions in the application based on the level of the importance of the role.

In this application, the latest web technologies have been used, such as *FuelPHP*, *Bootstrap* and the *Twig* framework. The web application offers a fully responsive design with the aim of being both simple and professional. It is based on the *MVC* software architecture (Model-View-Controller). The application allows a great degree of reuse of the code and facilitate the task of maintenance in the future.

The web application has been divided into three subsystems, performing for each of them all the phases of the software life cycle following a *waterfall model*.

Lastly, tests have been performed to ensure the compliance of the functional requirements and load and stress tests have been performed to ensure the robustness of the application developed in this bachelor thesis.

## Keywords

Bootstrap, Framework, FuelPHP, Management of a research group, Metadata, Model-View-Controller (MVC), Projects, Responsive, Twig, Waterfall Model, Web application.

## *Agradecimientos*

*En primer lugar, dar las gracias a mi tutor Simone Santini por darme la oportunidad de realizar este Trabajo de Fin de Grado y por su implicación y apoyo para lograr mi objetivo.*

*A todos mis amigos de la carrera con los que he compartido tiempo de estudio y diversión. Sin ellos no hubiera podido sacar adelante muchas asignaturas, porque gracias a su apoyo y ayuda se aprenden cosas que no se pueden aprender de un libro.*

*Por último, a mi familia por apoyarme cuando lo he necesitado y permitirme llegar hasta aquí superando todos los momentos difíciles que he pasado.*





# INDICE DE CONTENIDOS

<b>Glosario .....</b>	<b>v</b>
<b>1 Introducción.....</b>	<b>1</b>
1.1 Motivación.....	1
1.2 Objetivos.....	2
1.3 Organización de la memoria.....	3
<b>2 Estado del arte .....</b>	<b>5</b>
2.1 Soluciones Actuales del Problema .....	5
2.1.1 ActiveCollab <sup>(TM)</sup> .....	5
2.1.2 BaseCamp <sup>(TM)</sup> .....	7
2.1.3 Aceproject <sup>(TM)</sup> .....	8
2.1.4 Sumario comparativo.....	10
2.2 Tecnologías Utilizadas .....	11
2.2.1 Twig.....	11
2.2.2 Bootstrap.....	12
2.2.3 jQuery .....	13
2.2.4 AJAX.....	13
2.2.5 Modelo-Vista-Controlador .....	14
2.2.6 FuelPHP.....	14
<b>3 Diseño.....</b>	<b>17</b>
3.1 Ciclo de Vida.....	17
3.2 Alcance de la aplicación.....	18
3.3 Análisis de Requisitos .....	19
3.3.1 Requisitos Funcionales Subsistema Usuarios.....	19
3.3.2 Requisitos Funcionales Subsistema Proyectos.....	21
3.3.3 Requisitos Funcionales Subsistema de Estadísticas.....	25
3.3.4 Requisitos No Funcionales.....	26
3.4 Diagrama Casos de Uso.....	27
3.5 Diseño del Modelo de datos.....	28
3.6 Estructura del proyecto.....	30
<b>4 Desarrollo .....</b>	<b>33</b>
4.1 Extensión Twig, Modelos, Controladores y Enroutado .....	33
4.1.1 Extensión Twig.....	33
4.1.2 Modelos FuelPHP.....	34
4.1.3 Controladores y Enroutado .....	35
4.2 Desarrollo Subsistema Usuarios.....	36
4.2.1 Ejemplo Modelo Subsistema Usuarios.....	37
4.2.2 Ejemplo Controlador Subsistema Usuarios.....	38
4.3 Desarrollo Subsistema Proyectos .....	38
4.3.1 Ejemplo Manejador de Ficheros Subsistema Proyectos.....	39
4.4 Desarrollo Subsistema Estadísticas .....	39
4.4.1 Ejemplo Grafico Tipo de Recurso .....	40
<b>5 Integración, pruebas y resultados .....</b>	<b>41</b>
5.1 Pruebas Unitarias.....	41
5.2 Pruebas Funcionales .....	41
5.3 Pruebas Carga/Estrés .....	45
<b>6 Conclusiones y trabajo futuro .....</b>	<b>47</b>
6.1 Conclusiones.....	47
6.2 Trabajo futuro .....	47

<b>Referencias .....</b>	<b>49</b>
<b>Anexos.....</b>	<b>- 1 -</b>
A      Manual de instalación .....	- 1 -
B      Anexo Pruebas Funcionales.....	- 5 -

## INDICE DE FIGURAS

FIGURA 2-1: VISTA SEGUIMIENTO ACTIVECOLLAB .....	6
FIGURA 2-2: VISTA CALENDARIO ACTIVECOLLAB .....	6
FIGURA 2-3: VISTA CALENDARIO BASECAMP.....	7
FIGURA 2-4: VISTA VERSIONES BASECAMP .....	8
FIGURA 2-5: VISTA DIAGRAMA DE GANTT DE ACEPROJECT .....	9
FIGURA 2-6: INTERFAZ ACEPROJECT .....	9
FIGURA 2-7: EJEMPLO PLANTILLA TWIG .....	11
FIGURA 2-8: FORMULARIO CON BOOTSTRAP .....	12
FIGURA 2-9: CÓDIGO EJEMPLO JQUERY .....	13
FIGURA 2-10: CÓDIGO EJEMPLO AJAX.....	13
FIGURA 2-11: DIAGRAMA MVC.....	14
FIGURA 2-12: EJEMPLO CÓDIGO CONTROLADOR FUELPHP .....	14
FIGURA 2-13: EJEMPLO CÓDIGO ORM .....	15
FIGURA 3-1: CICLO DE VIDA EN CASCADA .....	17
FIGURA 3-2: DIAGRAMA DE CASOS DE USO 4RESEARCHERSOFT .....	27
FIGURA 3-3: MODELO ENTIDAD-RELACIÓN 4RESEARCHERSOFT.....	28
FIGURA 3-4: ESQUEMA RELACIONAL CON EJEMPLO 4RESEARCHERSOFT .....	30
FIGURA 4-1: EJEMPLO DEL CÓDIGO DE LA EXTENSIÓN TWIG DISEÑADA. ....	33
FIGURA 4-2: EJEMPLO USO DE LA EXTENSIÓN PERSONALIZADA EN LA PLANTILLA TWIG. ....	34
FIGURA 4-3: EJEMPLO TRADUCCIÓN DE RUTA PARA EL CALENDARIO DE 4RESEARCHERSOFT. ....	35
FIGURA 4-4: EJEMPLO CONTROLADOR PARA EL CALENDARIO DE 4RESEARCHERSOFT.....	36
FIGURA 4-5: DIAGRAMA CLASES ROLES 4RESEARCHERSOFT.....	36

FIGURA 4-6: EJEMPLO MODELO PERFIL DE USUARIO .....	37
FIGURA 4-7: EJEMPLO BORRADO USUARIO DESDE EL CONTROLADOR.....	38
FIGURA 4-8: PATRÓN DE DISEÑO SINGLETON .....	38
FIGURA 4-9: EJEMPLO DESCARGA DE UN RECURSO .....	39
FIGURA 4-10: EJEMPLO CÓDIGO ESTADÍSTICA GRAFICO TIPO DE RECURSO LADO SERVIDOR.....	40
FIGURA 4-11: EJEMPLO CÓDIGO ESTADÍSTICA GRAFICO TIPO DE RECURSO LADO CLIENTE .....	40
FIGURA 5-1: RESULTADOS JMeter FASE AUTENTICACIÓN .....	45
FIGURA 5-2: RESULTADOS JMeter FASE ADMINISTRACIÓN .....	45
FIGURA A-1: PANEL INSTALACIÓN XAMPP.....	- 1 -
FIGURA A-2: CREACIÓN BASE DE DATOS MySQL CON NAVICAT .....	- 2 -
FIGURA A-3: PROGRAMADOR DE TAREAS WINDOWS 7.....	- 4 -
FIGURA A-4: SCRIPT VINCULADO POR EL PROGRAMADOR DE TAREAS DE WINDOWS 7.....	- 4 -

## INDICE DE TABLAS

TABLA 2-1: SUMARIO COMPARATIVO.....	10
TABLA 2-2: ETIQUETAS TWIG .....	12
TABLA 2-3: COMPARACIÓN FRAMEWORKS ESTUDIADOS.....	16
TABLA 3-1: ESTRUCTURA EN CARPETAS 4RESEARCHERSOFT .....	31
TABLA 4-1: TABLA EXPLICATIVA MODELOS FUELPHP .....	35
TABLA 5-1: TABLA PRUEBA FUNCIONAL REQUISITO FUNCIONAL 2.....	41
TABLA 5-2: TABLA PRUEBA FUNCIONAL REQUISITO FUNCIONAL 3.....	41
TABLA 5-3: TABLA PRUEBA FUNCIONAL REQUISITO FUNCIONAL 10.....	42
TABLA 5-4: TABLA PRUEBA FUNCIONAL REQUISITO FUNCIONAL 11.....	42
TABLA 5-5: TABLA PRUEBA FUNCIONAL REQUISITO FUNCIONAL 12.....	42
TABLA 5-6: TABLA PRUEBA FUNCIONAL REQUISITO FUNCIONAL 15.....	43

TABLA 5-7: TABLA PRUEBA FUNCIONAL REQUISITO FUNCIONAL 19.....	43
TABLA 5-8: TABLA PRUEBA FUNCIONAL REQUISITO FUNCIONAL 24.....	43
TABLA 5-9: TABLA PRUEBA FUNCIONAL REQUISITO FUNCIONAL 26.....	44
TABLA 5-10: TABLA PRUEBA FUNCIONAL REQUISITO FUNCIONAL 27.....	44
TABLA 5-11: TABLA PRUEBA FUNCIONAL REQUISITO FUNCIONAL 31.....	44
TABLA B-1: TABLA PRUEBA FUNCIONAL REQUISITO FUNCIONAL 4.....	- 5 -
TABLA B-2: TABLA PRUEBA FUNCIONAL REQUISITO FUNCIONAL 5.....	- 5 -
TABLA B-3: TABLA PRUEBA FUNCIONAL REQUISITO FUNCIONAL 6.....	- 5 -
TABLA B-4: TABLA PRUEBA FUNCIONAL REQUISITO FUNCIONAL 7.....	- 6 -
TABLA B-5: TABLA PRUEBA FUNCIONAL REQUISITO FUNCIONAL 8.....	- 6 -
TABLA B-6: TABLA PRUEBA FUNCIONAL REQUISITO FUNCIONAL 9.....	- 6 -
TABLA B-7: TABLA PRUEBA FUNCIONAL REQUISITO FUNCIONAL 13.....	- 6 -
TABLA B-8: TABLA PRUEBA FUNCIONAL REQUISITO FUNCIONAL 14.....	- 7 -
TABLA B-9: TABLA PRUEBA FUNCIONAL REQUISITO FUNCIONAL 16.....	- 7 -
TABLA B-10: TABLA PRUEBA FUNCIONAL REQUISITO FUNCIONAL 17.....	- 7 -
TABLA B-11: TABLA PRUEBA FUNCIONAL REQUISITO FUNCIONAL 18.....	- 8 -
TABLA B-12: TABLA PRUEBA FUNCIONAL REQUISITO FUNCIONAL 20.....	- 8 -
TABLA B-13: TABLA PRUEBA FUNCIONAL REQUISITO FUNCIONAL 21.....	- 8 -
TABLA B-14: TABLA PRUEBA FUNCIONAL REQUISITO FUNCIONAL 22.....	- 8 -
TABLA B-15: TABLA PRUEBA FUNCIONAL REQUISITO FUNCIONAL 23.....	- 9 -
TABLA B-16: TABLA PRUEBA FUNCIONAL REQUISITO FUNCIONAL 25.....	- 9 -
TABLA B-17: TABLA PRUEBA FUNCIONAL REQUISITO FUNCIONAL 28.....	- 9 -
TABLA B-18: TABLA PRUEBA FUNCIONAL REQUISITO FUNCIONAL 29.....	- 9 -
TABLA B-19: TABLA PRUEBA FUNCIONAL REQUISITO FUNCIONAL 30.....	- 10 -
TABLA B-20: TABLA PRUEBA FUNCIONAL REQUISITO FUNCIONAL 32.....	- 10 -

## Glosario

---

<b>AJAX</b>	Acrónimo de Asynchronous JavaScript and XML, es una técnica de desarrollo web.
<b>Aplicación Web</b>	Herramienta por la cual los usuarios acceden a un servidor web a través de internet.
<b>Array</b>	Medio para guardar un conjunto de objetos del mismo tipo.
<b>Bootstrap</b>	Framework HTML, CSS y JavaScript.
<b>Cliente</b>	Aplicación informática que consume un servicio de otro ordenador.
<b>CSIC</b>	Siglas de Consejo Superior de Investigaciones Científicas.
<b>CSRF</b>	Cross-site request forgery. Es un ataque malicioso en el que comandos no autorizados son transmitidos por el usuario.
<b>CSS</b>	Cascading Style Sheets. Lenguaje para definir y crear estilos de un documento HTML.
<b>CSV</b>	Comma separated values. Tipo de documento abierto para representar datos.
<b>DOM</b>	Document Object Model. Conjunto de utilidades específicamente diseñadas para manipular documentos HTML.
<b>Framework</b>	Estructura tecnológica y de soporte basada en funcionalidades y módulos para ayudar a desarrollar un software.
<b>FuelPHP</b>	Framework basado en PHP para desarrollar una aplicación web.
<b>GRID</b>	Forma de representación de datos en tablas componiéndose de filas y columnas.
<b>HTML</b>	Hyper Text Markup Language. Lenguaje de marcado para la elaboración de páginas web basado en etiquetas.
<b>HTTP</b>	HyperText Transfer Protocol. Protocolo de comunicación en la web.
<b>JavaScript</b>	Lenguaje de programación débilmente tipado.
<b>JQuery</b>	Librería JavaScript creada para simplificar la manera de interactuar con los documentos HTML.
<b>JSON</b>	Acrónimo de JavaScript Object Notation. Formato de texto ligero para el intercambio de datos.

<b>Licencia BSD</b>	Licencia software permisiva para los sistemas Berkeley Software Distribution.
<b>Metadatos</b>	Datos que describen otros datos.
<b>MVC</b>	Modelo Vista Controlador. Patrón de arquitectura software que separa los datos y la lógica de una aplicación.
<b>MySQL</b>	Sistema de gestión de bases de datos relacional.
<b>PHP</b>	Hypertext Preprocessor. Lenguaje de programación del lado del servidor.
<b>Plugin</b>	Aplicación que se relaciona con otra para aportarle una funcionalidad nueva y específica.
<b>Responsive</b>	Es una técnica de diseño web para la correcta visualización de la misma página en distintos dispositivos.
<b>RESTful</b>	Representational State Transfer. Estilo de arquitectura software para el traspaso de información de la World Wide Web.
<b>Servidor</b>	Aplicación informática que atiende peticiones del cliente y devuelve una respuesta.
<b>Script</b>	Programa que realiza diversas tareas como interactuar con el sistema operativo.
<b>SQL</b>	Structured Query Language. Lenguaje declarativo de acceso a base de datos relacionales.
<b>SSL</b>	Secure Sockets Layer. Protocolo criptográfico para comunicaciones seguras en la red.
<b>URI</b>	Uniform resource identifier. Cadena de caracteres que identifica los recursos de una red de forma unívoca.
<b>URL</b>	Uniform Resource Locator. Identificador de recursos uniformes.
<b>XAMPP</b>	Servidor independiente de plataforma que consiste en la gestión de bases de datos, el servidor apache y PHP
<b>XML</b>	eXtensible Markup Language. Es un metalenguaje para definir lenguajes de marcas.
<b>XSS</b>	Tipo de inseguridad informática que permite a una persona inyectar código JavaScript.

# 1 Introducción

---

En el siguiente capítulo se detallará la motivación de este Trabajo de Fin de Grado explicando el alcance del proyecto, los objetivos de este y la organización de la memoria. Este trabajo pertenece al área de competencia tecnológica de Ingeniería del Software dentro de la Ingeniería Informática.

## 1.1 Motivación

Hoy en día las aplicaciones Web forman parte de nuestras vidas aumentando notablemente el impacto de la informática en nuestro día a día. Muchas actividades aprovechan las funcionalidades sofisticadas y la facilidad de acceso de las aplicaciones web como elemento facilitador. Concretamente en este Trabajo de Fin de Grado se busca facilitar la gestión de un grupo de investigación ayudando así a los investigadores a gestionar los proyectos en los que participan y el material producido en estos proyectos [1]. Una aplicación se diseña con el objetivo de integrarse en una actividad específica y de satisfacer ciertas necesidades de esta actividad. Este diseño en particular, empieza por tanto con una pregunta: ¿Qué es lo que un Investigador necesita para agilizar su trabajo?

Un punto clave de las colaboraciones es la comunicación entre los investigadores a través de la transmisión del material de los proyectos (Informes, artículos, etc...), y, gracias a la web, se dispondrá de este material al instante permitiendo a los investigadores aprovechar al máximo su tiempo y lograr así llegar al objetivo deseado de una forma más rápida. Además de lo mencionado anteriormente, una aplicación de este tipo es útil para la organización (Universidad, CSIC, Laboratorios Industriales, etc.) que gestiona el proyecto facilitando la gestión de todos sus investigadores y de sus colaboradores externos llevando de una forma más ordenada el seguimiento de un proyecto, sus participantes y su material, pudiendo así, organizar de una forma más adecuada sus recursos.

En el ámbito de gestión, la aplicación web gestionará los principales elementos de un grupo de investigación: los proyectos, miembros del grupo, el material de ese proyecto, el tipo del proyecto, las colaboraciones externas y los eventos que pueda tener.

Debido a las motivaciones expuestas anteriormente se ha decidido implementar una aplicación Web que facilite la tarea de gestión para mejorar el rendimiento de los investigadores. Dicha aplicación se desarrolla con las tecnologías más novedosas para mejorar la experiencia de usuario. A partir de este punto, se utilizará **4ResearchersSoft** como nombre de la aplicación Web.

## 1.2 Objetivos

El objetivo de este Trabajo de Fin de Grado es el diseño y la implementación de una aplicación Web que permita gestionar los principales aspectos de un grupo de Investigación. La aplicación desarrollada tendrá un carácter general, sin embargo, para fijar los detalles de la implementación se ha utilizado el ejemplo de los grupos de investigación de la Escuela Politécnica Superior de la Universidad Autónoma de Madrid, entorno en el que se ha llevado a cabo este trabajo. La aplicación Web permitirá:

- Dar un acceso cerrado. El sistema requerirá una autenticación, para ello se otorgará al administrador la capacidad de crear las cuentas para los investigadores. Cada investigador tendrá diferentes permisos dependiendo del rol que desempeñe en el proyecto pudiendo ser: ***Responsable del Proyecto, Miembro o Colaborador Externo***.
- Gestionar los investigadores, llevando un registro de la información importante de cada miembro o colaborador externo (experiencia del investigador, organismo del colaborador externo, etc.).
- Crear un modelo de datos basado en una base de datos que permitirá gestionar todos los aspectos del grupo de investigación y sus proyectos.
- Gestionar los Proyectos que se llevan a cabo almacenando los datos relevantes del proyecto, sus informes, sus artículos y sus datos con formato multimedia con metadatos. Estos proyectos tendrán un tipo específico para poder organizarlos de una manera más adecuada.
- Permitir crear Eventos y su tipo para organizar mejor el trabajo de cada investigador permitiendo así llevar un registro del calendario de sus proyectos. Estos tipos en principio son: ***Reuniones, Avisos e Hitos***. El administrador podrá añadir nuevos tipos y tanto el administrador de la aplicación como el responsable de cada proyecto podrán dar de alta a nuevos eventos.
- Gestionar los participantes de cada proyecto, tanto los nuevos participantes como los que abandonan el proyecto, permitiendo a los investigadores trabajar en diferentes proyectos de manera paralela.
- Almacenar en el servidor el material del proyecto desarrollado por el investigador con una localización propia para cada recurso.
- Llevar un registro de que investigador está trabajando con un recurso para bloquearlo a otros investigadores mientras se trabaja en él evitando así una edición en paralelo que traería problemas.
- Al ser una aplicación Web, tendrá como objetivo una interfaz de usuario intuitiva para agilizar las acciones del usuario de una forma adecuada.



### **1.3 Organización de la memoria**

La memoria está organizada en base a la normativa del Trabajo de Fin de Grado constando de los siguientes capítulos:

- **Capítulo 2 (Estado del Arte):** En este capítulo, se analizan las aplicaciones web de gestión de proyectos, se indican las nuevas tecnologías de programación web que se han empleado en este proyecto y el porqué de estas tecnologías.
- **Capítulo 3 (Diseño):** En este capítulo se define el alcance de la aplicación, su análisis, incorporando los requisitos funcionales y no funcionales de la aplicación, la metodología utilizada para el desarrollo de este proyecto y el diseño seguido para la creación del modelo de datos usado por la aplicación.
- **Capítulo 4 (Desarrollo):** En este capítulo se explica con detalle cómo se ha procedido para el desarrollo de la aplicación a través del framework FuelPHP explicando la lógica de la aplicación.
- **Capítulo 5 (Integración, pruebas y resultados):** En este capítulo se presentan las pruebas realizadas para demostrar la calidad del software y los resultados obtenidos en ellas.
- **Capítulo 6 (Conclusiones y trabajo futuro):** En este capítulo se exponen las conclusiones del Trabajo de Fin de grado y unas sugerencias para mejorar el trabajo.
- Por ultimo a estos capítulos mencionados anteriormente les sigue la bibliografía, el glosario del proyecto y los apéndices del documento.



## 2 Estado del arte

---

En este capítulo, se tratarán las diferentes opciones que hay en el mercado de la gestión de proyectos escogiendo las herramientas que más afinidad tengan con el problema planteado, explicando sus características y discutiendo sus aspectos positivos y negativos. Tras realizar esta comparación, se tratarán las tecnologías utilizadas en la aplicación desarrollada.

### 2.1 Soluciones Actuales del Problema

Hoy en día existen herramientas desarrolladas para resolver problemas similares a la gestión de un grupo de investigación como por ejemplo las herramientas de gestión de proyectos. En este apartado se compararán estas herramientas para analizar sus aspectos tanto positivos como negativos y la medida en que pueden ser adaptadas al problema.

Se analizarán las herramientas más completas en el aspecto de la gestión de proyectos online. Aunque estas herramientas son bastante completas, tienen limitaciones a la hora de definir todos los aspectos de un grupo de investigación. Las herramientas elegidas en esta comparación son: *ActiveCollab*, *BaseCamp* y *Aceproject*. Estas herramientas han sido elegidas entre otras candidatas por destacar en las prestaciones que ofrecen.

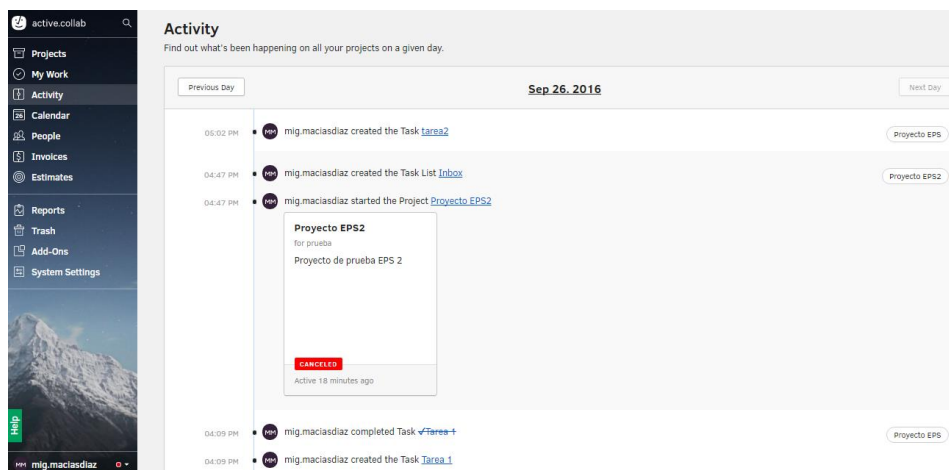
#### 2.1.1 ActiveCollab<sup>(TM)</sup>

ActiveCollab [2] es una aplicación web que permite gestionar y organizar proyectos grandes. Es una aplicación bastante completa por lo que muchas empresas la utilizan.

Esta aplicación web se encuentra disponible principalmente en inglés. Sin embargo, se pueden importar traducciones como archivos XML. Por el momento se permite importar varios idiomas como castellano, italiano o francés. El hecho que la aplicación permita el uso de varios idiomas es un aspecto positivo, sin embargo, la necesidad de tener que importarlos introduce una carga de trabajo notable para el usuario.

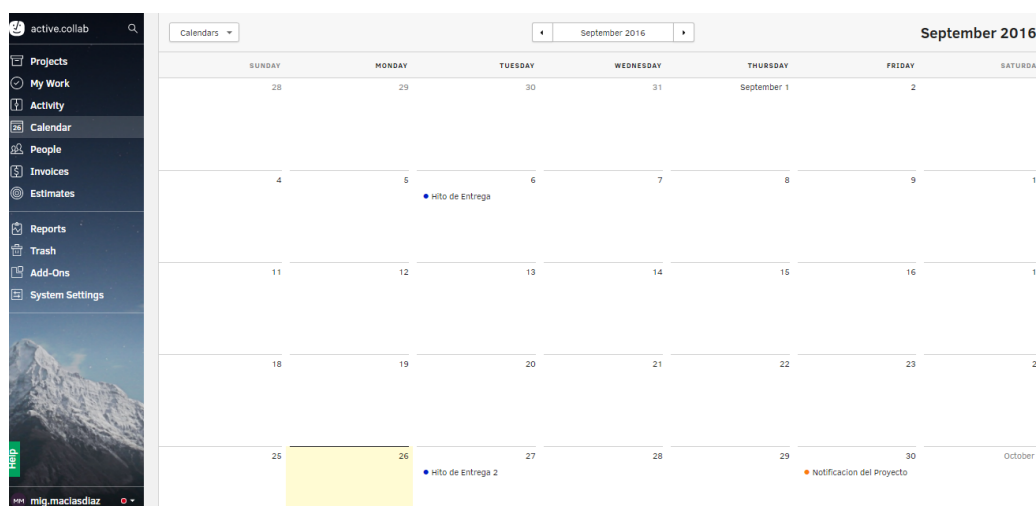
La aplicación permite la creación de múltiples proyectos y ofrece la posibilidad de añadir participantes de una forma intuitiva y eficaz, pero no incluye la capacidad de introducir un usuario temporal semejante a un colaborador externo (se debería borrar manualmente ese usuario del proyecto una vez terminada la colaboración). Un aspecto positivo de esta aplicación es la posibilidad de personalizar muchos aspectos que definen a un proyecto, permitiendo así adaptar la aplicación a muchos ámbitos diferentes. La aplicación permite subir ficheros a los proyectos, pero carece de un control de versiones o de la posibilidad de bloquear un fichero mientras se trabaja en él.

ActiveCollab está pensado para proyectos grandes ya que es posible dividir los proyectos largos en hitos facilitando así al usuario el seguimiento del proyecto. Otro punto positivo de ActiveCollab es la inclusión de una *vista* donde se puede mantener un seguimiento de la actividad de un proyecto en un día en concreto muy parecido al seguimiento que se hace en un sistema de control de versiones.



**Figura 2-1: Vista seguimiento ActiveCollab**

Por supuesto ActiveCollab también incluye la capacidad de filtrar elementos del proyecto como tareas, permitiendo así clasificar una tarea, añadir un hito e incluso dividir las tareas en subtareas. Estos filtros funcionan mediante reglas, las cuales, pueden ser definidas (por ejemplo, que miembro del equipo está trabajando en una tarea determinada). También incluye la vista en forma de calendario de nuestro proyecto.



**Figura 2-2: Vista calendario ActiveCollab**

Por último, ActiveCollab contiene funciones más sofisticadas como el control de tiempo, estimaciones de tiempo, facturación, presupuesto, una papelera donde se mantienen durante 30 días los proyectos que se borren y la posibilidad de conectarse con los archivos de Dropbox<sup>(TM)</sup> y Google Drive<sup>(TM)</sup>.

En conclusión, ActiveCollab es una aplicación excelente que incluye lo que se necesita para gestionar un proyecto grande, pero debido a que está diseñada para este tipo de proyectos, el usuario debe trabajar bastante para familiarizarse con ella y muchas de sus funciones resultaran superfluas para la gestión de proyectos relativamente pequeños como la mayoría de los proyectos de investigación. La aplicación no incluye la posibilidad de introducir un control en los ficheros del proyecto, no incluye la posibilidad de introducir un usuario temporal que trabaje como colaborador externo y no es un software libre ni gratuito.

### 2.1.2 BaseCamp<sup>(TM)</sup>

BaseCamp [3] es una aplicación web que permite gestionar proyectos de una forma colaborativa permitiendo a los usuarios estar al día en el proyecto que comparten. BaseCamp está pensado para proyectos cortos ya que si el proyecto se extiende en el tiempo su correcta organización puede resultar problemática.

La interfaz de BaseCamp es muy intuitiva. Un aspecto relevante es la inclusión de un chat para permitir a los miembros del proyecto mantener una conversación en tiempo real, lo cual facilita mucho el trabajo en equipo. La funcionalidad del chat es muy parecida a otra que permite gestionar mensajes a modo de hilos de conversación. Las diferencias principales entre estas funciones son dos:

- En el chat no se permite enviar adjuntos mientras que en el generador de mensajes sí.
- En el chat las conversaciones son en tiempo real mientras que en el generador no.

La función más importante de BaseCamp es, naturalmente, la gestión de proyectos. BaseCamp permite crear proyectos formados por sus miembros, asignar *responsabilidades* a cada miembro del proyecto y asignar tareas (*to-do* en Basecamp). Estas tareas, tendrán una fecha determinada y se incluirán en el calendario que BaseCamp facilita para administrar el proyecto.

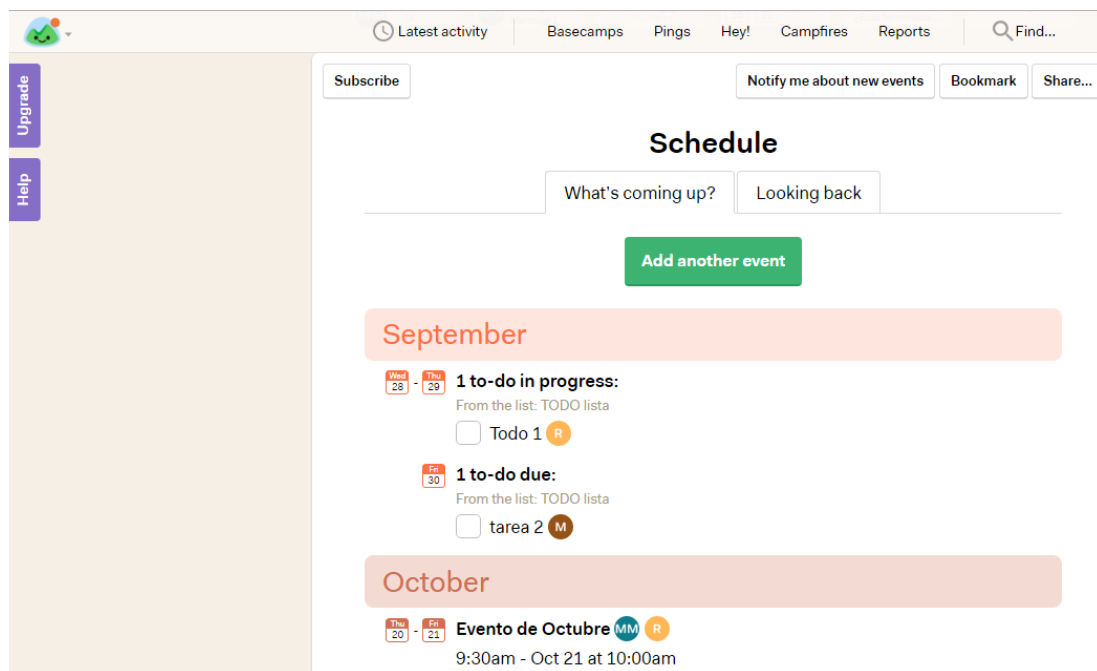
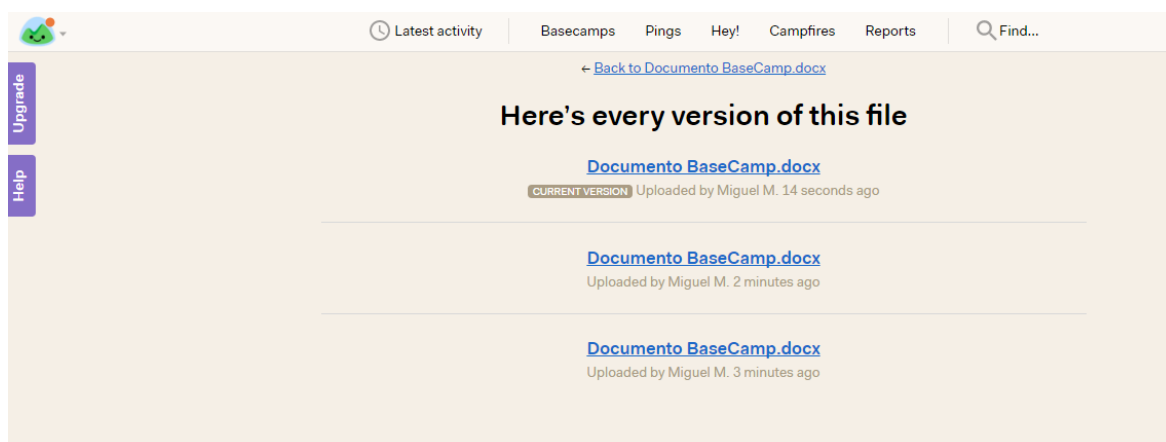


Figura 2-3: Vista calendario BaseCamp

BaseCamp incluye un registro de documentos o ficheros almacenándolos en la aplicación y permite crear documentos online. Hay una gran distinción entre almacenar un documento o un fichero. Los documentos creados online se pueden editar y mantienen un registro de los cambios. En cambio, los ficheros o documentos que son almacenados desde nuestro

ordenador mantienen un registro más completo ya que incorporan el registro anterior y aparte se puede volver a la versión anterior.



**Figura 2-4: Vista versiones BaseCamp**

Por último, BaseCamp permite el seguimiento del trabajo de cada miembro del proyecto mediante *Check-ins*, estos Check-ins son tareas automáticas que se envían a un miembro preguntando por ejemplo que ha realizado en el día de hoy. Un aspecto negativo de BaseCamp son los filtros que quizá sean demasiado sencillos y no permiten una selección muy sofisticada.

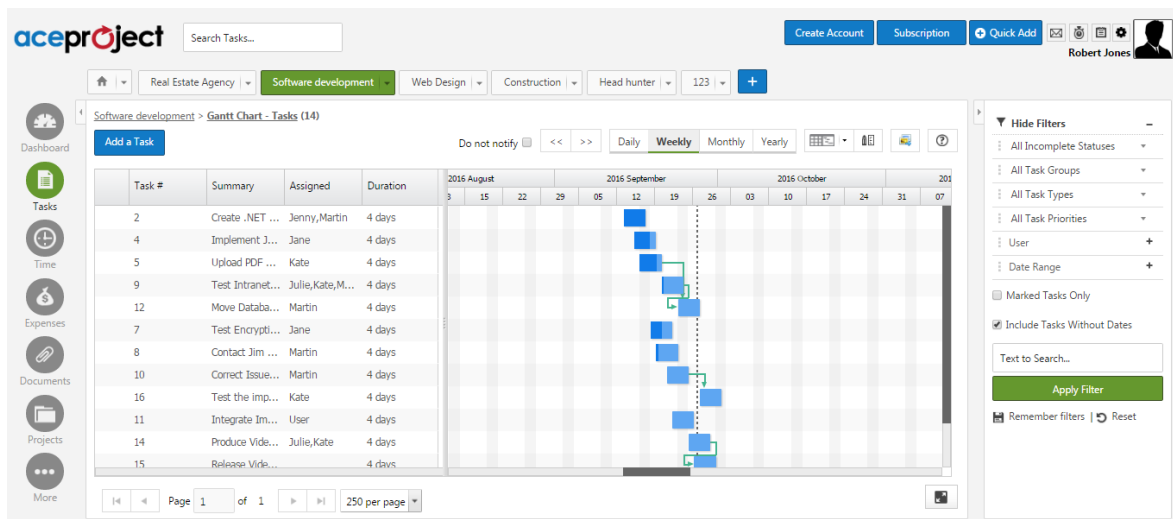
En conclusión, BaseCamp es una aplicación muy buena y sencilla para el uso centrada en proyectos cortos, permitiendo una buena comunicación entre los miembros del equipo. Tiene una gestión excelente de archivos y de calendario, pero si el proyecto es grande encontraremos problemas de gestión debido a la escasa flexibilidad de sus filtros.

### **2.1.3 Aceproject <sup>(TM)</sup>**

Aceproject [4] es una aplicación web que permite gestionar proyectos incluyendo: gestión de los recursos del proyecto, sus tareas y las prioridades entre tareas (utilizando diagramas de Gantt). Como ActiveCollab esta aplicación está pensada para proyectos muy grandes ya que incluye muchas opciones para la gestión de proyectos.

Aceproject es un software comercial, de pago. Es posible acceder a una demo online limitada (5 usuarios, 5 Proyectos, fuertes limitaciones en el almacenamiento y en las tareas). La evaluación de la aplicación que se realiza aquí, se basa en esta demo.

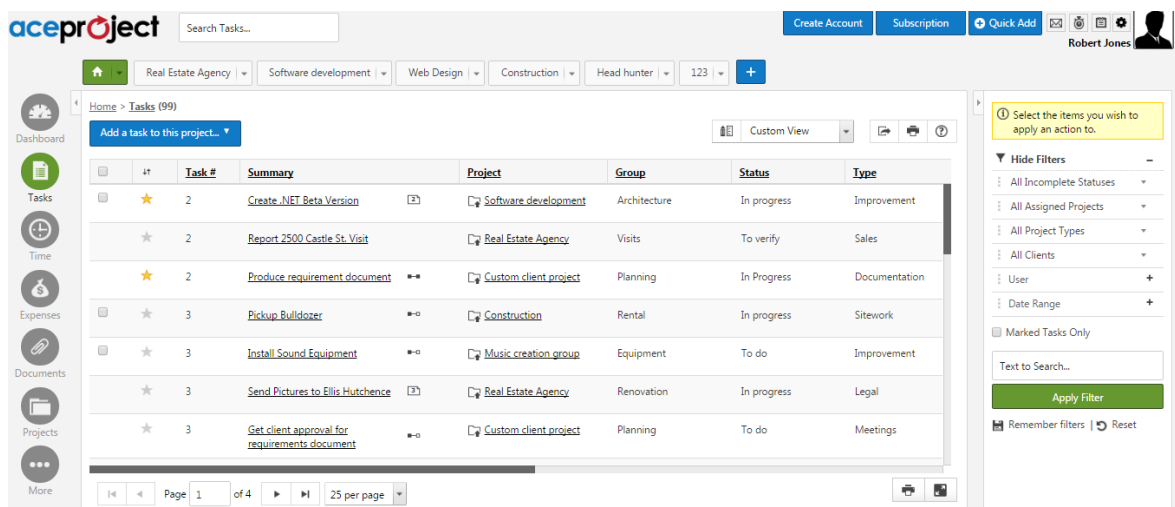
Así como las otras aplicaciones estudiadas, Aceproject permite la creación de proyectos, la creación de usuarios y la asignación de usuarios a proyectos. Aceproject incluye la posibilidad de definir roles de usuario permitiendo así que el mismo usuario pueda tener roles distintos en proyectos distintos. Aceproject también incluye tareas con prioridades facilitando así la gestión de los recursos. Lo más destacable de la gestión de proyectos es la posibilidad de definir diagramas de Gantt.



**Figura 2-5: Vista Diagrama de Gantt de Aceproject**

La gestión de documentos en Aceproject es muy completa ya que permite compartir documentos y tareas en los proyectos. Aceproject posibilita que los documentos sean bloqueados evitando así posibles conflictos y puede hacerlos públicos al cliente.

Por último, Aceproject permite administrar el presupuesto del proyecto permitiendo generar estadísticas de gastos. Por otro lado, su interfaz resulta a veces demasiado compleja y poco intuitiva (muy completa, pero su uso se hace muy tedioso).






**Figura 2-6: Interfaz Aceproject**

En conclusión, Aceproject es una aplicación muy completa en su versión de pago, pero muy limitada en la gratuita, se centra en proyectos largos permitiendo una buena gestión tanto de calendario como de archivos.

### 2.1.4 Sumario comparativo

Las aplicaciones analizadas en esta sección tienen objetivos muy similares a nuestros objetivos para gestionar un grupo de investigación. La mayoría de estas aplicaciones presentan una interfaz relativamente compleja y con una curva de aprendizaje bastante dura debido a la gran cantidad de servicios que ofrecen, a excepción de la aplicación BaseCamp que ofrece una interfaz muy intuitiva. A continuación, se muestra una tabla resumen de las aplicaciones estudiadas:

	 active.collab	 Basecamp®	 aceproject
<b>Facilidad Navegación e Interacción</b>	Difícil, es necesario mucho tiempo para familiarizarse con la aplicación.	Fácil, contenido muy bien expresado, intuitivo y se tarda poco en aprender a usarla.	Difícil, es necesario mucho tiempo para familiarizarse con la aplicación y la interfaz no es muy intuitiva
<b>Tipo proyectos a los que va dirigido</b>	Proyectos Grandes	Proyectos Cortos	Proyectos Grandes, pero también se adapta a cortos.
<b>Calidad gestión ficheros</b>	Mala, no permite una correcta gestión de los ficheros, ni versiones ni bloquea el acceso a ellos.	Buena, lleva un control de versiones muy bueno excepto para los documentos creados online.	Muy Buena, permite bloquear ficheros y lleva un buen control de ellos.
<b>Calidad organización de los proyectos</b>	Excelente, permite un seguimiento en forma de calendario muy bueno, un seguimiento del proyecto muy completo.	Malo, permite llevar una organización correcta, pero si el proyecto se hace demasiado grande se tendrán problemas con la organización.	Muy bueno, permite realizar diagramas de Gantt, incluye un calendario para las tareas.
<b>Comunicación entre miembros del equipo</b>	Buena, permite contactar con otros miembros del equipo.	Excelente, permite un chat en tiempo real e incluso un seguimiento del proyecto con mensajes.	Buena, permite la mensajería entre el equipo del proyecto.

**Tabla 2-1: Sumario Comparativo**

Gracias a este estudio, se ha podido determinar los puntos clave para mejorar la aplicación a desarrollar, dando importancia a la sencillez de la interfaz para que los usuarios puedan administrar sus proyectos y material del proyecto de forma sencilla y profesional.



## 2.2 Tecnologías Utilizadas

En este punto se lleva a cabo un análisis de las tecnologías que se han utilizado y de las razones que nos han llevado a su utilización. Se exponen los lenguajes modernos de desarrollo web y el patrón de arquitectura seguido en nuestra aplicación: Modelo-Vista-Controlador (MVC).

### 2.2.1 Twig

Twig es un motor de *plantillas* para PHP, basado en las plantillas Django [5] cuyo objetivo es facilitar la creación de vistas para las aplicaciones que utilizan el patrón de arquitectura MVC. Es capaz de generar *plantillas* con un código limpio y de fácil comprensión. Una plantilla es un archivo de texto que retorna formatos como HTML y CSV, teniendo variables y expresiones de control tales como bucles. Además, Twig introduce un grado de seguridad a nuestras vistas previniendo *ataques XSS*.

Otra característica de Twig es su flexibilidad ya que por defecto existen muchas *etiquetas* (palabras reservadas tales como *for*, *if*, *etc.*) que pueden ser utilizadas por el usuario, pero también ofrece la posibilidad de crear una extensión para definir nuestras propias etiquetas. Twig se distribuye bajo *licencia BSD* y es utilizado por muchos desarrolladores que utilizan frameworks como *Symfony*, *CodeIgniter* o *FuelPHP*.

```
<!DOCTYPE html>
<html>
  <head>
    <title>My Webpage</title>
  </head>
  <body>
    <ul id="navigation">
      {% for item in navigation %}
        <li><a href="{{ item.href }}">{{ item.caption }}</a></li>
      {% endfor %}
    </ul>

    <h1>My Webpage</h1>
    {{ a_variable }}
  </body>
</html>
```

Figura 2-7: Ejemplo Plantilla Twig

La figura 2-7 muestra un ejemplo de plantilla Twig, que evidencia como Twig ayuda a generar un código muy legible a pesar de la presencia de elementos como variables, bucles, etc. El ejemplo anterior es una plantilla muy sencilla, pero Twig también incluye opciones más sofisticadas, como la inclusión de otra plantilla o la herencia, permitiendo crear un esqueleto para nuestras plantillas y mediante la herencia ir especializando las plantillas.

A continuación, se muestra un pequeño resumen de las etiquetas más importantes en Twig:

Etiqueta	Descripción	Ejemplo
<b>for</b>	Recorre cada elemento, puede funcionar como el for o el foreach de PHP.	<code>{% for user in users %}</code> <code>  {{ user.email }}</code> <code>{% endfor %}</code>
<b>if</b>	Equivalente al if en PHP.	<code>{% if upper %}</code> <code>  texto a mayusculas.</code> <code>{% endif %}</code>
<b>include</b>	Inserta una plantilla devolviendo todo su contenido	<code>{% include 'user.twig' %}</code>
<b>extends</b>	Se utiliza para extender una plantilla	<code>{% extends 'user.twig' %}</code>
<b>filter</b>	Permite aplicar filtros Twig.	<code>{% filter user %}</code> <code>  {{ user.email }}</code> <code>{% endfilter %}</code>
<b>set</b>	Asigna un valor a una variable	<code>{% set name = 'Miguel' %}</code>

**Tabla 2-2: Etiquetas Twig**

En resumen, Twig es un motor de plantillas que permite crear vistas para el patrón MVC y destaca por su flexibilidad, su seguridad y por permitir el uso de herencia en las plantillas. Debido a estas razones se ha seleccionado Twig para este proyecto.

### 2.2.2 Bootstrap

Es un framework HTML, CSS y JavaScript [6] desarrollado por Twitter <sup>(TM)</sup> para diseñar aplicaciones web de una forma *responsive* (adaptable a diferentes dispositivos). Este framework se basa en un mecanismo *GRID* (basado en 12 columnas) que permite adaptarse a cualquier tipo de dispositivo (Móvil, Tablet, Ordenador de Escritorio...).

Bootstrap incluye entre otros elementos: formularios, botones, tablas. Es compatible con la mayoría de los navegadores (Google Chrome, Internet Explorer, Opera, etc...) y debido a esta compatibilidad se ha decidido utilizar este framework. La forma de utilizar este framework es por ejemplo otorgar una clase a un elemento HTML como se observa en el siguiente ejemplo:

☐ Remember me

```

<form class="form-inline">
  <div class="form-group">
    <label class="sr-only" for="exampleInputEmail3">Email address</label>
    <input type="email" class="form-control" id="exampleInputEmail3" placeholder="Enter email">
  </div>
  <div class="form-group">
    <label class="sr-only" for="exampleInputPassword3">Password</label>
    <input type="password" class="form-control" id="exampleInputPassword3" placeholder="Password">
  </div>
  <div class="form-check">
    <label class="form-check-label">
      <input class="form-check-input" type="checkbox"> Remember me
    </label>
  </div>
  <button type="submit" class="btn btn-primary">Sign in</button>
</form>

```

**Figura 2-8: Formulario con Bootstrap**

### 2.2.3 jQuery

Es una biblioteca de JavaScript [7] que ayuda a incorporar animaciones, modificaciones de comportamiento, eventos y validación de usuario. La biblioteca permite manipular el árbol DOM y utilizar AJAX facilitando así la interacción cliente-servidor. Existen varias versiones de la biblioteca actualmente activas. Se ha decidido utilizar la versión jQuery 2.2.4 que es compatible con los navegadores más importantes del mercado.

A continuación, se muestra un código jQuery de ejemplo del proyecto desarrollado que se encarga de añadir un evento para ocultar una ventana de confirmación:

```
Admin_Base.prototype.add_event_hide_modal = function (id_modal) {  
    $('#'+id_modal).on('show.bs.modal', function() {  
        var modal = $(this);  
        clearTimeout(modal.data('hideInterval'));  
        modal.data('hideInterval', setTimeout(function () {  
            modal.modal('hide');  
        }, 2000));  
    });  
}
```

Figura 2-9: Código ejemplo jQuery

### 2.2.4 AJAX

Es una técnica de desarrollo web para crear aplicaciones interactivas [8] permitiendo desde el cliente mantener una comunicación asíncrona con el servidor para recargar una página o para recibir una respuesta ya sea en formato JSON, XML o texto plano.

A continuación, se muestra un código AJAX de ejemplo del proyecto desarrollado que se encarga de enviar una petición al servidor y recibir una respuesta en formato JSON, con la cual se realizan operaciones sobre un elemento HTML:

```
$.ajax({  
    type: "GET",  
    url: '/admin/projects/'+this.value+'/users',  
    contentType: "application/json",  
    success: function (response) {  
        if (!response) {  
            $("#left_items").find('option').remove();  
            $("#left_items").append('<option class="list-group-item" value=0>No hay elementos</option>');  
            return;  
        }  
        var response_json = JSON.parse(response);  
        $("#left_items").find('option').remove();  
  
        if (Object.keys(response_json).length == 0) {  
            $("#left_items").append('<option class="list-group-item" value=0>No hay elementos</option>');  
            return;  
        }  
        response_json.forEach(function (user) {  
            $("#left_items").append('<option class="list-group-item" value=' + user['id'] + '>' + user['email'] + '</option>');  
        });  
    },  
});
```

Figura 2-10: Código ejemplo AJAX

## 2.2.5 Modelo-Vista-Controlador

El Esquema Modelo-Vista-Controlador es un patrón de arquitectura de software que separa los datos y la lógica de una aplicación de la interfaz de usuario [9]. Esto facilita que la aplicación sea más fácil de mantener y su escalabilidad más sencilla. Se compone de tres componentes:

- El **modelo** es la lógica de la aplicación, es decir, la representación de la información y como acceder a ella.
- La **vista** es la encargada de presentar el modelo de forma visual.
- El **controlador** responde a las peticiones de un usuario sobre el modelo enviando a la vista la información necesaria para que presente el modelo de una forma adecuada.

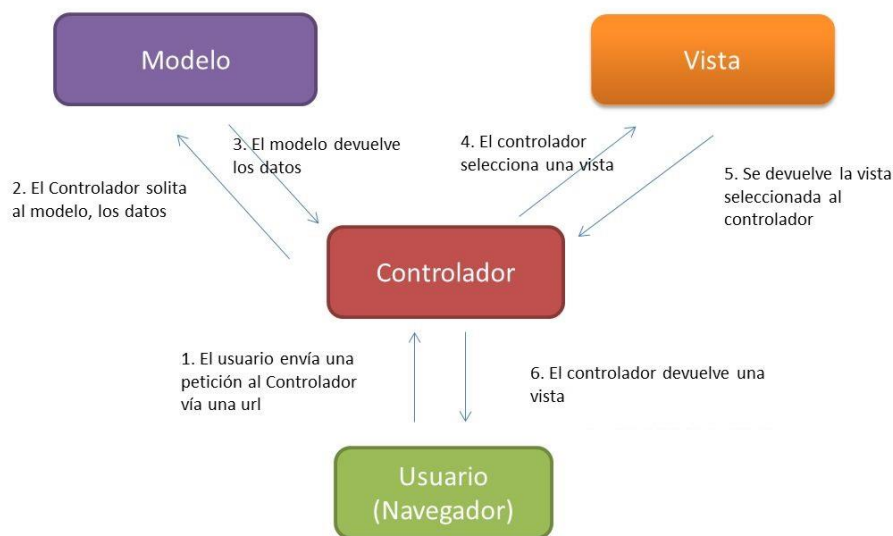


Figura 2-11: Diagrama MVC

## 2.2.6 FuelPHP

FuelPHP [10] es un framework desarrollado en el lenguaje de programación PHP y basado en el patrón de arquitectura Modelo-Vista-Controlador que está diseñado para tener soporte HMVC [11]. Sus principales diseñadores fueron Harro Vertin, Dan Horrigan, Philip Sturgeon, Frank de Jonge and Jelme Schreuder. Este framework requiere una versión de PHP superior a la 5.3 y la instalación de **composer**. Composer es un manejador de dependencias y su instalación se realiza de manera local. Para instalar FuelPHP siga el manual de instalación que se encuentra al final de este documento.

```
class Controller_Example extends Controller
{
    public function action_index()
    {
        $data['css'] = Asset::css(array('reset.css', '960.css', 'main.css'));
        return Response::forge(View::forge('welcome/index'));
    }
}
```

Figura 2-12: Ejemplo Código controlador FuelPHP

FuelPHP es el framework en que se ha implementado este proyecto. Se han tomado varias consideraciones para su elección:

FuelPHP tiene una sintaxis muy sencilla y es muy portable. Frameworks como Symfony son muy pesados y para aplicaciones medianas o pequeñas tienen demasiadas desventajas ya que están pensados para proyectos más grandes. FuelPHP es más flexible y, a pesar de poderse usar para proyectos grandes, se adapta sin carga excesiva en un proyecto pequeño. Otro framework estudiado ha sido laravel, el cual es un framework muy potente y tiene aspectos muy positivos como FuelPHP, pero, dada la facilidad de instalación y ligereza de FuelPHP se ha optado por el uso de FuelPHP.

FuelPHP incluye el uso de **POO** (Programación Orientada a Objetos) con su modelo **ORM** (Object Relational Mapper) que realiza dos tareas importantes, mapear las tablas de la base de datos y establecer las relaciones entre las tablas, siguiendo el Patrón Active record [12]. Este patrón se encarga de todas las operaciones de modificación y consulta en las tablas de nuestra base de datos. Este patrón trabaja sobre las tablas de la base de datos como si fueran objetos, sin por esto impedir la utilización de SQL si se considera necesario.

```
// Find only some columns
Model_Article::query()->select('id', 'name');
Model_Article::find('all', array('select' => array('id', 'name')));




// Find all columns except 'name'
Model_Article::query()->select(array('name' => false));
Model_Article::find('all', array('select' => array(array('name' => false))));

// include related models in the find
Model_Article::query()->related(array('author', 'comments'));
Model_Article::find('all', array('related' => array('author', 'comments')));
```

**Figura 2-13: Ejemplo Código ORM**

FuelPHP permite la internacionalización de la aplicación con su librería de idiomas, los objetos ORM mencionados anteriormente, la posibilidad de inclusión de motores de plantillas como Twig, Smarty. FuelPHP también incluye librerías útiles como la librería de subida de archivos, un sistema de cache, implementación **RESTful** etc.

A continuación, se muestra una comparación sobre los frameworks estudiados:

		 Symfony	 laravel
<b>Patrón de Diseño</b>	MVC	MVC	MVC
<b>Multilenguaje</b>	Si	Si	Si
<b>Ligereza</b>	Muy Ligero	Pesado	Ligero
<b>ORM</b>	Si	Si	Si
<b>Paradigma de programación soportados</b>	Orientado a objetos y funcional	Orientados a objetos y programación dirigida por eventos.	Orientados a objetos, funcional y programación dirigida por eventos.
<b>LDAP</b>	No	Si	Si
<b>Prevención Inyección SQL</b>	Si	Si	Si
<b>Facilidad de instalación</b>	Muy fácil	Fácil	Muy fácil

**Tabla 2-3: Comparación frameworks estudiados**

## 3 Diseño

---

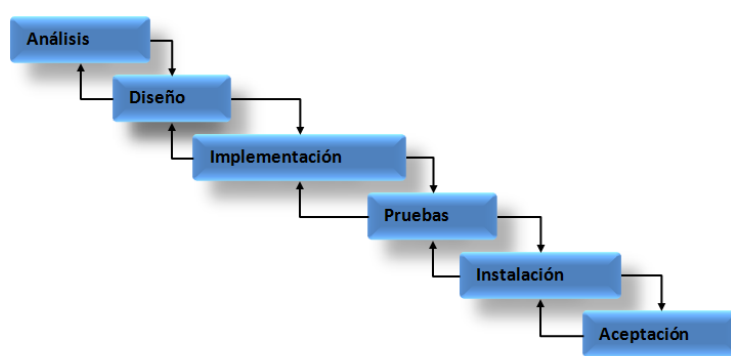
En este capítulo se tratarán los distintos aspectos del diseño de la aplicación. En particular, se discutirá el porqué del ciclo de vida seguido, el alcance de la aplicación, se continuará con la definición de los requisitos funcionales y no funcionales que la aplicación debe cumplir y el diseño seguido, incluyendo el diagrama de casos de usos, el modelo de datos y la estructura en carpetas de la aplicación 4ResearchersSoft.

La primera fase de 4ResearchersSoft, consistió en la investigación de las tecnologías más punteras mencionadas anteriormente y en el análisis de aplicaciones similares. La segunda fase de investigación consistió en investigar sobre el framework FuelPHP para tener conocimientos de lo que es capaz de hacer y así poder definir con más precisión los requisitos funcionales de la aplicación. Tras las fases de investigación, se ha decidido dividir la aplicación en tres subsistemas. El *Subsistema Usuarios* que se encarga de todo lo relacionado con los usuarios y sus roles, el *Subsistema Proyectos y Metadatos* que se encarga de todo lo relacionado con los proyectos, su material y los metadatos del material. Y, por último, el *Subsistema de Estadísticas*, que se encarga de todas las estadísticas que se generan.

### 3.1 Ciclo de Vida.

Para la realización de la aplicación 4ResearchersSoft se ha seguido un modelo en cascada [13] el cual es un modelo clásico utilizado con frecuencia y su pilar es seguir en orden las fases del proyecto permitiendo así una metodología de trabajo efectiva.

En este modelo, cuando se completa una fase, se puede volver a la fase anterior en caso de ser necesario. Este modelo tiene como inconvenientes que cualquier error de diseño detectado nos hace retornar a esa fase incrementando el coste del desarrollo y también otro inconveniente es que hasta que no se complete una fase no se puede llevar a cabo la siguiente fase.



Ciclo de vida en cascada

**Figura 3-1: Ciclo de vida en cascada**

Se ha elegido el método en cascada debido a que antes de comenzar cualquier tipo de desarrollo se tiene un plan y una visión clara. Esto lleva a que se pueda iniciar la aplicación con bastante rapidez estimando los plazos de una forma más precisa.

### **3.2 Alcance de la aplicación.**

El objetivo principal de la aplicación es facilitar la gestión de proyectos de un grupo de investigación mediante la gestión de los participantes del proyecto, las colaboraciones externas existentes y por último almacenar el material del proyecto permitiendo una gestión de los artículos, informes y archivos multimedia que permita que la gestión del material no entre en conflicto. Como ya se ha mencionado anteriormente, la aplicación se compone de tres subsistemas principales.

En primer lugar, el ***Subsistema Usuarios*** permitirá a los usuarios de la aplicación realizar las siguientes tareas:

- Entrar en el sistema y permitir a cada usuario actualizar su perfil.
- Recuperar la contraseña de un usuario mediante el envío de un correo electrónico.
- Cambiar la contraseña de un usuario desde su perfil.
- Distinguirá cuatro tipos de usuarios, Administrador, Responsable del Proyecto, Miembro y Colaborador Externo.
- El colaborador externo será un usuario temporal definido por el administrador.
- El administrador podrá cambiar el rol de un usuario (solo habrá un administrador por lo tanto este rol es reservado).
- El administrador podrá añadir nuevos organismos de colaboración externa.
- Tendrá en cuenta el grupo al que pertenece el usuario para realizar las tareas mencionadas anteriormente.

En segundo lugar, el ***Subsistema Proyectos y Metadatos*** permitirá realizar las siguientes tareas:

- Gestionar la localización del material de cada proyecto en el servidor.
- Dar de alta y actualizar los diferentes proyectos.
- Dar de alta, de baja y actualizar a los miembros de un proyecto.
- Cerrar un proyecto cuando se haya completado.
- Dar de alta, de baja y actualizar los eventos de un proyecto.
- Descargar y poder subir al servidor el material del proyecto (informes, artículos...etc.).
- Bloquear la descarga del material descargado del proyecto evitando conflictos.
- Guardar los metadatos asociados a cada elemento del material del proyecto.
- Añadir nuevos tipos tanto para los eventos como para los proyectos.
- Añadir los organismos de los colaboradores externos.
- Eliminar el material del proyecto.

Por último, el ***Subsistema de Estadísticas*** permitirá realizar las siguientes tareas:

- Incluir un gráfico que indicará el numero activo de recursos por mes.
- Incluir un gráfico que indicará el porcentaje de un tipo de material en concreto en el servidor.



### 3.3 Análisis de Requisitos

En esta sección se exponen los requisitos tanto funcionales como no funcionales de la aplicación divididos en sus respectivos subsistemas mostrando los datos de entrada y de salida correspondientes a cada requisito.

#### 3.3.1 Requisitos Funcionales Subsistema Usuarios.

##### (RF1) Roles Usuario

Descripción: La aplicación distinguirá cuatro tipos de roles de usuario:

- I. **Administrador:** Podrá realizar todas las tareas mencionadas en el apartado 3.2.
- II. **Responsable de Proyecto:** Podrá dar de alta y de baja a los eventos del proyecto en los que sea el responsable y también se le permitirá realizar las mismas tareas que a un usuario con el rol de Miembro.
- III. **Miembro:** Se le permitirá actualizar su perfil, subir material del proyecto, descargar el material del proyecto y visualizar su calendario de trabajo.
- IV. **Colaborador Externo:** Realizará las mismas tareas que un miembro, pero este usuario será un usuario temporal, donde el tiempo de estancia vendrá definido por el administrador.

##### (RF2) Registro de Usuario

Descripción: La aplicación permitirá al administrador registrar nuevos usuarios desde su panel de administración.

**Entrada:** El administrador deberá introducir como datos obligatorios el email del usuario, contraseña, rol al que pertenece (exceptuando el rol de administrador), nombre y apellidos del usuario. En caso de que el rol de usuario sea colaborador externo también contarán como datos obligatorios el organismo procedente del colaborador y el tiempo de estancia en la aplicación.

**Proceso:** El sistema comprobará los datos introducidos y si son correctos registrará al usuario en una base de datos.

**Salida:** El sistema redirigirá al usuario a una pantalla de inicio desde la que se podrá acceder a las funciones reservadas a dicho usuario, creando automáticamente un perfil básico con los datos introducidos.

##### (RF3) Login de Usuario en la Aplicación.

Descripción: La aplicación permitirá al usuario el entrar en la aplicación.

**Entrada:** El usuario deberá introducir su email y contraseña.

**Proceso:** El sistema comprobará los datos introducidos en la base de datos y si son correctos, el usuario entrará en el sistema.

**Salida:** Si el proceso es correcto el sistema redirigirá al usuario a una pantalla de inicio desde la que se podrá acceder a las funciones reservadas a dicho usuario, si no el sistema redirigirá al usuario a la pantalla de login.

#### *(RF4) Recuperación contraseña Usuario.*

Descripción: La aplicación permitirá al usuario recuperar su contraseña si es incapaz de entrar a su cuenta enviándole un email con una contraseña nueva.

**Entrada:** El usuario en cuestión deberá introducir su email.

**Proceso:** El sistema comprobará la existencia de ese usuario (con su email) y enviará automáticamente un email a la cuenta vinculada con su nueva contraseña generada.

**Salida:** El email con la nueva contraseña generada, se cambiará la contraseña en la base de datos y se informará con un mensaje de éxito o de error en caso de fallo.

#### *(RF5) Actualización información perfil de Usuario.*

Descripción: La aplicación permitirá al usuario cambiar los datos de su perfil en estos datos se incluyen: nombre, apellidos, fecha nacimiento, experiencia y contraseña.

**Entrada:** El usuario en cuestión deberá introducir los cambios en los datos mencionados en la descripción.

**Proceso:** El sistema comprobará los datos introducidos y si es correcto se cambiará la información en la base de datos modificando solo aquella que haya sido afectada.

**Salida:** El sistema informará si el cambio se ha realizado con éxito o informando del error en caso de fallo en los introducidos.

#### *(RF6) Dar de baja a un Usuario en la aplicación.*

Descripción: La aplicación permitirá al administrador dar de baja a un usuario desde su panel de administración.

**Entrada:** Identificador del usuario a dar de baja.

**Proceso:** Se eliminará al usuario comprobando que no esté participando en ningún proyecto.

**Salida:** El sistema informará si se ha dado de baja con éxito o ha surgido algún error. Si no ha ocurrido ningún error se eliminará al usuario de la base de datos.

#### *(RF7) Editar un Usuario.*

Descripción: La aplicación permitirá al administrador editar los datos de un usuario como su perfil y su rol (El rol administrador no podrá cambiar ya que solo habrá un administrador).

**Entrada:** Los datos de perfil o rol del usuario.

**Proceso:** Se cambiará los datos afectados comprobando que no hay ningún conflicto en ellos.

**Salida:** El sistema informará si se ha realizado el cambio con éxito, mensaje de error si ha ocurrido algún error.

#### *(RF8) Dar de alta a Organismos.*

Descripción: La aplicación permitirá dar de alta nuevos organismos.

**Entrada:** El administrador deberá introducir el nombre del organismo.

**Proceso:** El sistema comprobará que no existe ese organismo y lo insertará en la base de datos.

**Salida:** El sistema informará si se ha realizado con éxito o con un mensaje de error si ha ocurrido algún error.

#### *(RF9) Dar de baja a Organismos.*

Descripción: La aplicación permitirá dar de baja a los organismos.

**Entrada:** Identificador del organismo a dar de baja.

**Proceso:** El sistema comprobará que ese organismo no tiene ningún colaborador externo activo y lo dará de baja.

**Salida:** El sistema informará si se ha realizado con éxito o con un mensaje de error si ha ocurrido algún error.

#### *(RF10) Búsqueda de Usuarios.*

Descripción: La aplicación permitirá al administrador filtrar a los usuarios por email, nombre, fecha de nacimiento y por el rol al que pertenecen.

**Entrada:** Cualquiera de los datos mencionados en la descripción.

**Proceso:** El sistema comprobará si existen usuarios en la base de datos con los requisitos del filtro si los hay devolverá estos usuarios.

**Salida:** Lista de usuarios que cumplen los requisitos del filtro mostrándolos por pantalla.

### **3.3.2 Requisitos Funcionales Subsistema Proyectos.**

#### *(RF11) Dar de Alta a un Proyecto.*

Descripción: La aplicación permitirá al administrador dar de alta a un proyecto.

**Entrada:** El administrador deberá introducir el nombre del proyecto y su tipo como datos obligatorios. También podrá especificar el responsable del proyecto, sus participantes, su descripción y el coste estimado del proyecto.

**Proceso:** El sistema comprobará los datos introducidos y si son correctos registrará el proyecto en una base de datos.

**Salida:** El sistema informará si se ha realizado con éxito o con un mensaje de error si ha ocurrido algún error. Si no ha ocurrido ningún error se añadirá a la base de datos y se redirigirá a la vista de gestión de proyectos.

#### *(RF12) Cerrar Proyecto.*

Descripción: La aplicación permitirá al administrador cerrar un proyecto.

**Entrada:** Identificador del proyecto a cerrar.

**Proceso:** El sistema cerrará el proyecto y bloqueará toda acción sobre ese proyecto a excepción de la visualización de su material, eventos pasados y características.

**Salida:** El sistema informará si se ha realizado con éxito o con un mensaje de error si ha ocurrido algún error. Si no ha ocurrido ningún error se borrará el proyecto.

#### *(RF13) Dar de alta Tipos de proyectos.*

Descripción: La aplicación permitirá dar de alta tipos de proyectos.

**Entrada:** El administrador deberá introducir el nombre del tipo de proyecto.

**Proceso:** El sistema comprobará que no existe ese tipo de proyecto y lo insertará en la base de datos.

**Salida:** El sistema informará si se ha realizado con éxito o con un mensaje de error si ha ocurrido algún error.

#### *(RF14) Dar de baja Tipos de proyectos.*

Descripción: La aplicación permitirá dar de baja tipos de proyectos.

**Entrada:** Identificador del tipo de proyecto a dar de baja.

**Proceso:** El sistema comprobará que ese tipo de proyecto no tiene ningún proyecto asociado y lo dará de baja.

**Salida:** El sistema informará si se ha realizado con éxito o con un mensaje de error si ha ocurrido algún error.

#### *(RF15) Dar de Alta a un Usuario en un proyecto.*

Descripción: La aplicación permitirá al administrador dar de alta a los usuarios en un proyecto determinado desde su panel de administración.

**Entrada:** El administrador deberá introducir el email del usuario a dar de alta en el proyecto seleccionado.

**Proceso:** El sistema registrará al usuario como participante del proyecto.

**Salida:** El sistema informará si el registro en el proyecto del usuario se ha realizado correctamente o error en caso contrario.

#### *(RF16) Dar de baja a un Usuario en un proyecto.*

Descripción: La aplicación permitirá al administrador dar de baja a los usuarios de un proyecto desde su panel de administración.

**Entrada:** Identificador del usuario a dar de baja del proyecto.

**Proceso:** Se desligará al usuario con el proyecto en cuestión.

**Salida:** El sistema informará si se ha dado de baja con éxito o ha surgido algún error.

### *(RF17) Localización del material del Proyecto.*

Descripción: La aplicación seguirá el siguiente recorrido para guardar el material del proyecto: fuel/app/library/proyecto\_id/files/File donde:

- fuel/app: es la ruta del framework FuelPHP que indica la aplicación.
- proyecto\_id: Id del proyecto que contiene el material.
- File: nombre del material.

### *(RF18) Actualizar información de un Proyecto.*

Descripción: La aplicación permitirá al administrador cambiar la información de un proyecto.

**Entrada:** El administrador introducirá los cambios en los datos del proyecto.

**Proceso:** El sistema comprobará los datos introducidos y si es correcto se cambiará la información del proyecto en la base de datos.

**Salida:** El sistema informará si el cambio se ha realizado con éxito o informando del error en caso de fallo en los datos introducidos.

### *(RF19) Dar de alta un evento en un Proyecto.*

Descripción: La aplicación permitirá al administrador y al responsable del Proyecto dar de alta nuevos eventos.

**Entrada:** El administrador o al responsable del proyecto deberá introducir el nombre del evento, el tipo del evento, los participantes del evento y la fecha del evento.

**Proceso:** El sistema comprobará los datos introducidos y añadirá al proyecto el evento creado.

**Salida:** El sistema informará si se ha dado de alta el evento con éxito o informando del error en caso de fallo en los introducidos.

### *(RF20) Dar de baja un evento en un Proyecto.*

Descripción: La aplicación permitirá al administrador eliminar un evento en el proyecto correspondiente.

**Entrada:** Identificador del evento a dar de baja del proyecto.

**Proceso:** Se eliminará el evento correspondiente en el proyecto.

**Salida:** El sistema informará si se ha eliminado el evento correctamente. Si no ha ocurrido ningún error se eliminará de la base de datos.

### *(RF21) Modificar información de un evento.*

Descripción: La aplicación permitirá al administrador y al responsable del proyecto modificar el evento.

**Entrada:** El administrador o al responsable del proyecto introducirá los datos a modificar.

**Proceso:** El sistema comprobará los datos introducidos y si son correctos modificará la información del evento.

**Salida:** El sistema informará si la modificación del evento se ha realizado correctamente o error en caso contrario.

### *(RF22) Dar de alta Tipos de eventos.*

Descripción: La aplicación permitirá dar de alta tipos de eventos.

**Entrada:** El administrador deberá introducir el nombre del tipo de evento.

**Proceso:** El sistema comprobará que no existe ese tipo de evento y lo insertará en la base de datos.

**Salida:** El sistema informará si se ha realizado con éxito o con un mensaje de error si ha ocurrido algún error.

### *(RF23) Dar de baja Tipos de eventos.*

Descripción: La aplicación permitirá dar de baja tipos de eventos.

**Entrada:** Identificador del tipo de evento a dar de baja.

**Proceso:** El sistema comprobará que ese tipo de evento no tiene ningún evento asociado y lo dará de baja.

**Salida:** El sistema informará si se ha realizado con éxito o con un mensaje de error si ha ocurrido algún error.

### *(RF24) Almacenar en el servidor material del proyecto.*

Descripción: La aplicación permitirá al usuario almacenar en el servidor cualquier tipo de fichero (Informe, formato multimedia, etc.) en el proyecto seleccionado en el que participe.

**Entrada:** Fichero que se quiera almacenar.

**Proceso:** Se almacenará en el servidor dicho fichero en la ruta correspondiente (Mirar RF13), si este fichero existiese y se encontrase bloqueado se quitaría el bloqueo de descarga.

**Salida:** El sistema informará si se ha realizado el cambio con éxito, mensaje de error si ha ocurrido algún error.

### *(RF25) Eliminar material del proyecto.*

Descripción: La aplicación permitirá al administrador eliminar material del proyecto correspondiente.

**Entrada:** Identificador del recurso que se quiera eliminar.

**Proceso:** Se eliminará el fichero elegido por el administrador para ser eliminado.

**Salida:** El sistema informará si se ha dado eliminado con éxito o ha surgido algún error.

### *(RF26) Descarga material del proyecto.*

Descripción: La aplicación permitirá al usuario descargar el material del proyecto en el que esté trabajando.

**Entrada:** Ninguno.

**Proceso:** Se descargará el fichero seleccionado y se bloqueará las futuras descargas del fichero para evitar conflictos en la edición en paralelo.

**Salida:** Fichero seleccionado.

#### *(RF27) Añadir metadatos al material del proyecto.*

Descripción: La aplicación asociará metadatos automáticamente al material del proyecto, estos datos serán: fecha última modificación, título del archivo, versión y el formato del archivo.

**Entrada:** Ninguna.

**Proceso:** Ninguno.

**Salida:** Ninguna.

#### *(RF28) Búsqueda de Proyectos.*

Descripción: La aplicación permitirá al administrador la búsqueda de proyectos por nombre, coste estimado y su tipo.

**Entrada:** Atributos mencionados en la descripción por los que se quieran filtrar.

**Proceso:** El sistema buscará los proyectos que coincidan con las especificaciones y la listará.

**Salida:** El sistema mostrará la lista de los proyectos que cumplen las condiciones.

#### *(RF29) Búsqueda de eventos.*

Descripción: La aplicación permitirá al administrador la búsqueda de eventos por nombre, tipo de evento, proyecto al que pertenece, fecha y hora.

**Entrada:** Atributos mencionados en la descripción por los que se quieran filtrar.

**Proceso:** El sistema buscará los eventos que coincidan con las especificaciones y los listará.

**Salida:** El sistema mostrará la lista de los eventos que cumplen las condiciones.

#### *(RF30) Búsqueda de Material de Proyecto por tipo.*

Descripción: La aplicación permitirá al administrador la búsqueda del material de un proyecto por nombre y tipo de fichero.

**Entrada:** Atributos mencionados en la descripción por los que se quieran filtrar.

**Proceso:** El sistema buscará los ficheros del proyecto que coincidan con las especificaciones y la listará.

**Salida:** El sistema mostrará la lista de los ficheros del proyecto que cumplen las condiciones.

### **3.3.3 Requisitos Funcionales Subsistema de Estadísticas.**

#### *(RF31) Grafico Número de Recursos Activos por mes.*

Descripción: La aplicación permitirá al administrador ver en un mes determinado cuantos archivos hay activos.

**Entrada:** Recursos activos y mes para generar el grafico.

**Proceso:** El sistema comprobará cuando se ha trabajado en un recurso del proyecto y lo contabilizará.

**Salida:** El sistema mostrará el grafico resumen de los recursos activos.

### *(RF32) Grafico Tipo de material.*

Descripción: La aplicación permitirá al administrador visualizar un gráfico con el porcentaje de un tipo de material en concreto que hay en el servidor.

**Entrada:** Material que comprobar.

**Proceso:** El sistema calculará los porcentajes asociados.

**Salida:** El sistema mostrará el grafico por pantalla.

### **3.3.4 Requisitos No Funcionales.**

A continuación, se muestran los requisitos no funcionales del sistema.

#### *(RNF1) Requisitos de Disponibilidad.*

(RNF 1.1) Se realizarán mantenimientos y durante estos mantenimientos la aplicación no estará disponible.

#### *(RNF2) Requisitos de Seguridad.*

(RNF 2.1) La aplicación exigirá contraseñas seguras (mínimo 8 caracteres alfanuméricos combinando mayúsculas y minúsculas).

(RNF 2.2) La contraseña se encriptará con un algoritmo de encriptación para evitar robo de identidad.

(RNF 2.3) El sistema dispondrá de una cadena CSRF para evitar los ataques CSRF.

#### *(RNF3) Requisitos de Usabilidad.*

(RNF 3.1) La aplicación tendrá una respuesta inferior a 30 segundos, si se supera este tiempo, la petición no se procesará.

(RNF 3.2) En caso de fallo, la aplicación deberá mostrar un mensaje de error informando de lo ocurrido.

(RNF 3.3) La aplicación utilizará tecnologías AJAX para mejorar el tiempo de respuesta.

#### *(RNF4) Requisitos de Backup.*

(RNF 4.1) La aplicación realizará una copia de seguridad de la base de datos para evitar pérdidas de información.

#### *(RNF5) Requisitos de Acceso.*

(RNF 5.1) La aplicación estará disponible tanto en castellano como en inglés, a excepción de la validación del lado del cliente que estará disponible solamente en inglés.

#### *(RNF6) Requisitos de Plataformas.*

(RNF 6.1) La aplicación será compatible con los navegadores web Firefox, Opera, Internet Explorer (Versión 9 o superior) y Google Chrome.



(RNF 6.2) La aplicación será instalada en un servidor Web con el Sistema Operativo Windows 7 con el servidor independiente XAMPP.

(RNF 6.3) La aplicación utilizará el gestor de base de datos MySQL.

### (RNF7) Requisitos de Espacio.

(RNF 7.1) La aplicación permitirá la subida de ficheros al servidor de como máximo 100 Megabytes.

## 3.4 Diagrama Casos de Uso.

Un *diagrama de casos de uso* define los casos de uso en la que los usuarios operan con la aplicación [14]. A continuación, se muestra el diagrama de casos de uso de la aplicación 4ResearchersSoft:

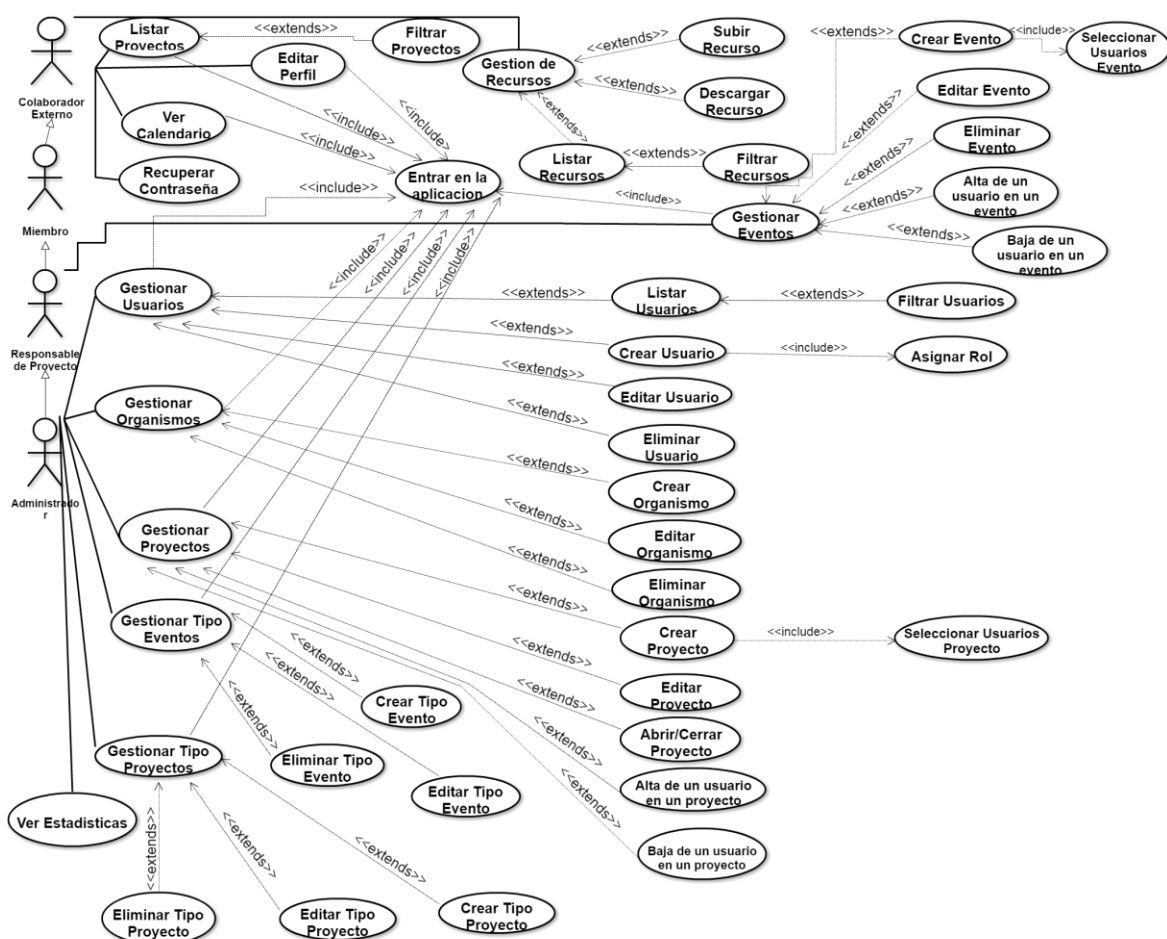
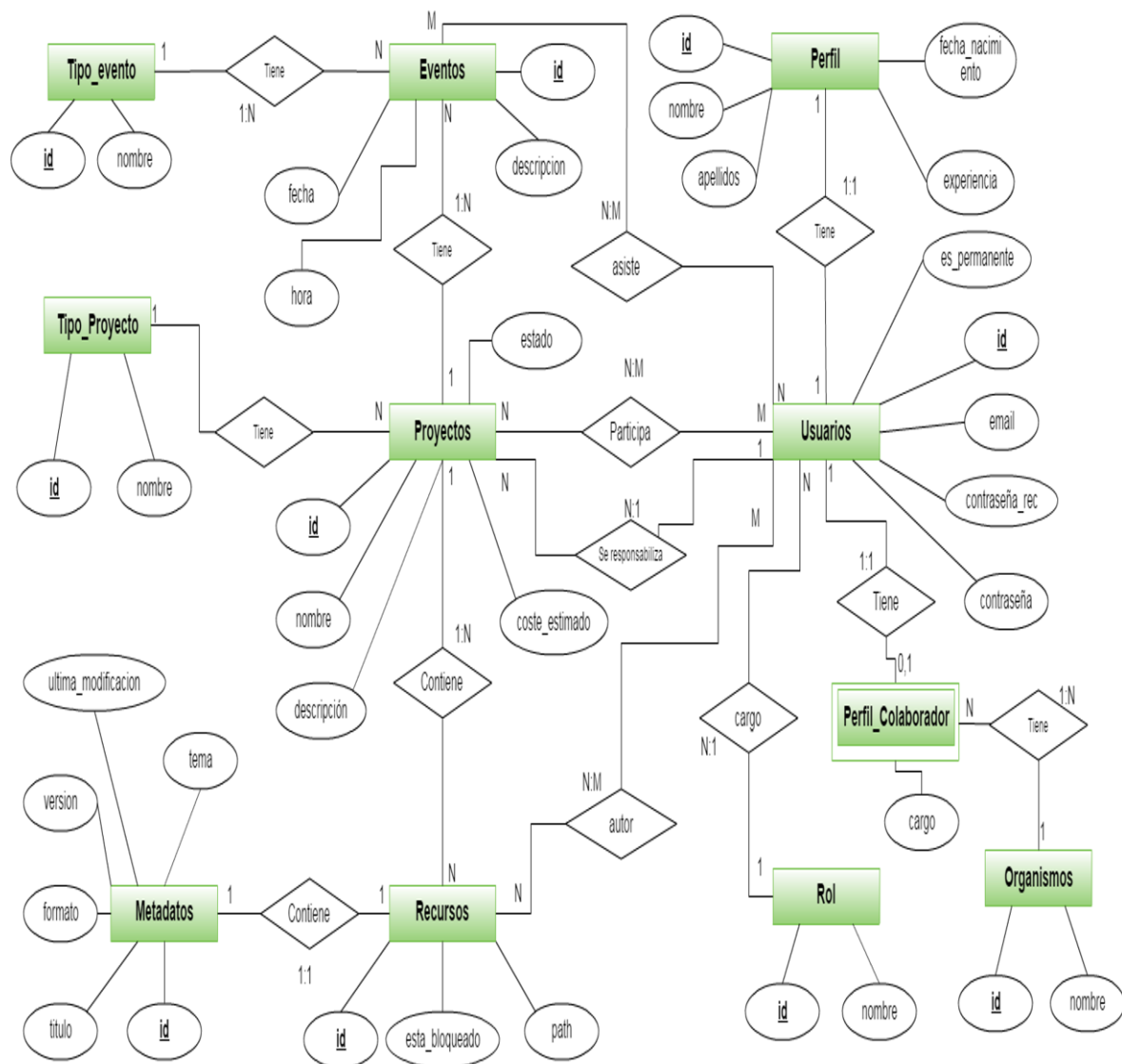


Figura 3-2: Diagrama de Casos de Uso 4ResearchersSoft

Como se observa en la Figura 3-2 se puede observar los cuatro tipos de roles que tendrá la aplicación y las tareas que pueden realizar. Se puede observar que un Colaborador Externo y un Miembro realizan las mismas tareas, solo difiere en que el Colaborador Externo es un usuario temporal. Por otro lado, el Responsable del Proyecto podrá gestionar eventos (solo aquellos de los que sea responsable). Y por último se puede observar que el Administrador puede realizar todas las tareas de la aplicación.

### 3.5 Diseño del Modelo de datos.

4ResearchersSoft utiliza una base de datos relacional para el modelo de datos, La figura 3-3 ilustra el modelo entidad relación [15] de la base de datos.



**Figura 3-3: Modelo Entidad-Relación 4ResearchersSoft**

En la Figura 3-3 no se muestran las entidades que permiten a la aplicación el uso de varios idiomas para evitar complicar el entendimiento del diagrama entidad-relación ya que estas entidades no aportan ningún tipo de información. Para conseguir que la aplicación sea totalmente multilinguaje en español e inglés se han estudiados dos técnicas:

- Añadir en cada tabla que necesite traducción, una columna por cada traducción necesaria.
- Crear una tabla de idiomas de la aplicación, y por cada tabla que necesite traducción crear una tabla adicional con todas sus traducciones y relacionarlas [16].

Se ha optado por la segunda opción ya que evita tener demasiados campos vacíos y es más flexible si en un futuro se quiere añadir un nuevo idioma.

A continuación, se describen las entidades del modelo entidad-relación:

- **Usuarios:** Es la entidad que representa a los usuarios de la aplicación. Cada usuario debe tener asociados un perfil (en el caso de un colaborador externo también tendrá asociado un perfil de colaborador) y un rol. Cada usuario se relaciona con los proyectos y eventos en los que participa. También se relaciona con los recursos de los que son autores.
- **Perfil:** Es la entidad que representa el perfil del usuario.
- **Perfil Colaborador:** Es una entidad débil que representa los datos del colaborador, es dependiente de la entidad usuarios y muestra los datos adicionales del colaborador como el cargo o su organismo.
- **Rol:** Es la entidad que representa los tipos de roles de la aplicación en este caso contendrá los roles: Administrador, Responsable del Proyecto, Miembro y Colaborador Externo.
- **Proyectos:** Es la entidad que gestiona los proyectos. Estos proyectos se relacionan con los eventos, recursos y usuarios. Se relaciona con la entidad tipo de proyecto para determinar su tipo.
- **Eventos:** Es la entidad que gestiona los eventos y sus datos. Estos eventos se relacionan con el proyecto al que pertenece el evento, los usuarios asistentes al evento y el tipo del evento.
- **Tipo proyecto:** Es la entidad que representa los tipos de proyectos existentes en la aplicación.
- **Tipo evento:** Es la entidad que representa los tipos de eventos existentes en la aplicación.
- **Organismos:** Es la entidad que representa los organismos de los que pueden provenir los colaboradores externos.
- **Recursos:** Es la entidad que gestiona los recursos de un proyecto. Estos recursos están asociados al proyecto al que pertenece, sus autores y a los metadatos del recurso.
- **Metadatos:** Es la entidad que gestiona los metadatos del recurso, tales como su formato y versión.

A continuación, tras explicar el modelo Entidad-Relación de 4ResearchersSoft se muestra el esquema relacional con un ejemplo sencillo representativo:

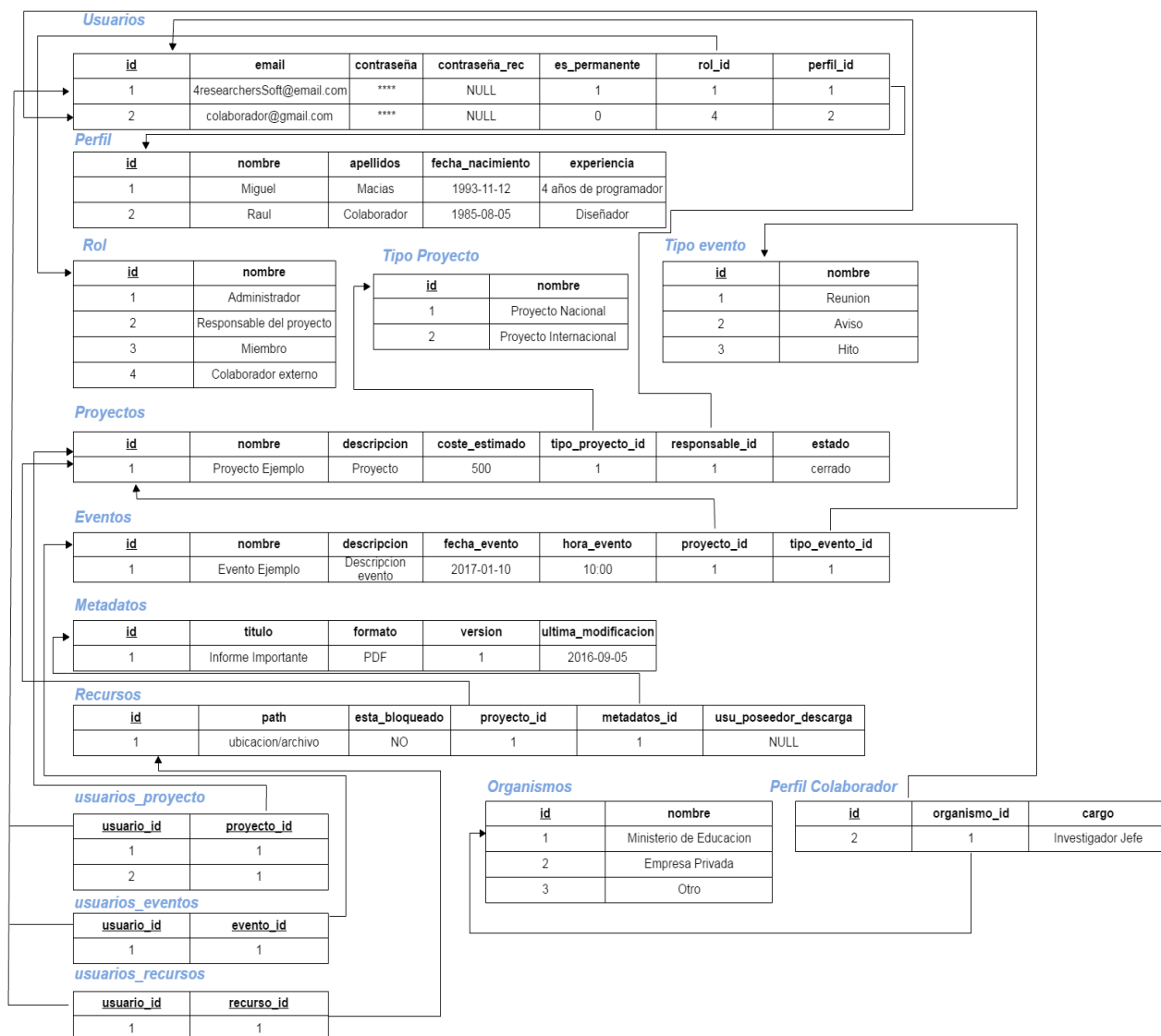


Figura 3-4: Esquema Relacional con ejemplo 4ResearchersSoft

### 3.6 Estructura del proyecto

En esta sección se describe la estructura de 4ResearchersSoft con sus componentes más importantes para facilitar la comprensión de cómo está estructurada la aplicación. (Ver Tabla 3-1).

Se debe tener en cuenta esta estructura para trabajar correctamente con FuelPHP y la aplicación. La carpeta más importante para el desarrollador es la carpeta **app** ya que en esta carpeta se almacena la aplicación. En esta carpeta se localizan los controladores, los modelos, las vistas, los archivos de configuración y las tareas de la aplicación. En la carpeta de configuración se encuentra el fichero **routes.php** el cual guarda todas las rutas de la aplicación.

Otra carpeta importante es la carpeta **public** que contendrá las hojas de estilo, los plugins de la aplicación y todos los archivos necesarios del lado del cliente.

En la carpeta raíz de la aplicación se localiza el archivo **composer.json** el cual contiene la información básica de la aplicación y permite incluir librerías instalables.

<b>Carpeta</b>	<b>Descripción</b>
<b>Database Backup</b>	Esta carpeta incluye todos los backups de la base de datos.
<b>public</b>	Contiene las hojas de estilo, JavaScript, <i>plugins</i> , imágenes e index.php.
<b>fuel</b>	Contiene las carpetas app, core, packages y vendor
<b>fuel/app</b>	Contiene todo lo relacionado con la aplicación como los controladores, vistas, modelos, tareas, configuración etc.
<b>fuel/core</b>	Contiene el código del framework (no se debe editar para evitar problemas en futuras actualizaciones)
<b>fuel/packages</b>	Contiene librerías de terceros y tus propias librerías para organizar mejor tu código.
<b>fuel/vendor</b>	Componentes instalados de terceros como puede ser Twig.
<b>fuel/app/config</b>	Contiene los archivos de configuración de la aplicación como las rutas.
<b>fuel/app/lang</b>	Contiene los diferentes idiomas de la aplicación con sus respectivos archivos php que contienen las traducciones.
<b>fuel/app/tasks</b>	Contiene las tareas de la aplicación
<b>fuel/app/views</b>	Contiene las vistas de la aplicación.
<b>fuel/app/classes/controller</b>	Contiene los controladores de la aplicación.
<b>fuel/app/classes/model</b>	Contiene los modelos de la aplicación.
<b>fuel/app/clases/twig</b>	Contiene la extensión de Twig diseñada.

**Tabla 3-1: Estructura en carpetas 4ResearchersSoft**

Por último, es recomendable no editar la carpeta **core** ya que incluye el código del framework y cualquier **upgrade** de versión del framework sobrescribiría el código, por lo que si se quiere extender código del framework este código deberá incluirse en la carpeta **app**.



## 4 Desarrollo

En este capítulo se exponen los detalles relativos a la fase de desarrollo, que corresponde con la implementación y documentación de la aplicación. Como se ha comentado anteriormente la aplicación se realizará bajo el framework FuelPHP, utilizando Mysql como gestor de base de datos y las plantillas Twig para las vistas. Para realizar esta etapa de desarrollo se ha optado por el IDE [17] JetBrains PhpStorm [18] ya que soporta todas las características modernas y previene errores con su tecnología, además, se adapta al trabajo de mucho de los frameworks existentes como Laravel y FuelPHP. Por ultimo este capítulo se dividirá en cuatro subsecciones, la primera hablará del funcionamiento de la extensión de Twig desarrollada, el funcionamiento de los modelos en FuelPHP y la relación entre el enroutado y los controladores en FuelPHP. Las subsecciones restantes trataran cada subsistema mencionado anteriormente en la parte de diseño de este documento.

### 4.1 Extensión Twig, Modelos, Controladores y Enroutado

En esta subsección se detalla la extensión Twig desarrollada, los modelos en FuelPHP y se explica cómo funciona el enroutado en FuelPHP.

#### 4.1.1 Extensión Twig

En esta aplicación se ha desarrollado una extensión Twig propia para agregar apoyo a la internalización de la aplicación y el uso de funciones del propio framework como etiquetas propias de Twig. Una extensión en Twig permite una mejor separación entre el código que se ejecuta en tiempo de compilación y el código en tiempo de ejecución, por tanto, mejora la velocidad de la aplicación. A continuación, se muestra una imagen en la que se muestra una parte del código de la extensión diseñada.

```
public function getFunctions()
{
    return array(

        'count'          => new Twig_Function_Function('count'),
        'min'            => new Twig_Function_Function('min'),
        'sha1'           => new Twig_Function_Function('sha1'),
        'var_dump'       => new Twig_Function_Function('var_dump'),
        'input_get'       => new Twig_Function_Function('Input::get'),
        'input_post'     => new Twig_Function_Function('Input::post'),
        'input_param'    => new Twig_Function_Function('Input::param'),
        'input_ajax'     => new Twig_Function_Function('Input::is_ajax'),
        'fuel_version'   => new Twig_Function_Method($this, 'fuel_version'),
        'fuel_env'       => new Twig_Function_Method($this, 'fuel_environment'),
        'config'         => new Twig_Function_Function('Config::get'),
        'lang'           => new Twig_Function_Method($this, 'auxiliarLanguage'),
        'html_anchor'    => new Twig_Function_Function('Html::anchor'),
        'asset_img'       => new Twig_Function_Function('Asset::img'),
        'asset_render'   => new Twig_Function_Function('Asset::render'),
        'session_get_flash' => new Twig_Function_Function('Session::get_flash'),
        'session_get'    => new Twig_Function_Function('Session::get'),
        'asset_css'      => new Twig_Function_Function('Asset::css'),
        'asset_js'       => new Twig_Function_Function('Asset::js'),
        'generate_csrf'  => new Twig_Function_Method($this, 'auxiliarTokenCsrf'),
        'lang_code'      => new Twig_Function_Function('Fuel\\Core\\Lang::get_lang'),
        'rand'           => new Twig_Function_Function('mt_rand'),
        'is_admin'       => new Twig_Function_Function('Auth\\Auth::is_admin'),
        'is_manager'     => new Twig_Function_Function('Auth\\Auth::is_manager'),
        'uri_link'       => new Twig_Function_Method($this, 'uri_link'),
        'get_user_id'    => new Twig_Function_Method($this, 'auxiliarGetUserId'),
    );
}
```

Figura 4-1: Ejemplo del código de la extensión Twig diseñada.

En la figura 4-1 se muestran funciones que luego se utilizarán en las plantillas Twig. Por ejemplo, se define una etiqueta `session_get` la cual tendrá la funcionalidad de la función `session:get_flash` de FuelPHP. Con esto permitimos aprovecharnos de la funcionalidad de FuelPHP en las plantillas Twig.

```
<div class="container">
  <span class="text-danger">{{ session_get_flash("message:error") }}</span>
</div>
<div class="container">
  <span class="text-success">{{ session_get_flash("message:ok") }}</span>
</div>
```

**Figura 4-2: Ejemplo uso de la extensión personalizada en la plantilla Twig.**

La figura 4-2 muestra cómo se puede utilizar las etiquetas creadas en la extensión Twig desde la propia plantilla.

### 4.1.2 Modelos FuelPHP

Los modelos en FuelPHP se utilizan para recuperar, manipular o eliminar datos de nuestro modelo de datos, es decir son una representación de estos datos. Los modelos en FuelPHP se relacionan con el modelo de datos a través del ORM que permite utilizar las tablas de la base de datos como objetos, para ello hay que definir las relaciones. Estas relaciones se definen de la siguiente manera:

Relación	Explicación	Ejemplo
<b>Belongs To</b>	Esta relación se utiliza tanto para relaciones <b>1-N</b> o <b>1-1</b> . Se utiliza cuando la clave primaria de la relación se encuentra en su tabla.	<pre>protected static \$_belongs_to = array(     'profile' =&gt; array(         'key_from' =&gt; 'profile_id',         'key_to' =&gt; 'id',         'model_to' =&gt;         'Auth\\Model_Auth_Profile',         'cascade_delete' =&gt; true     ), );</pre>
<b>Has One</b>	Esta relación se utiliza en una relación <b>1-1</b> y no incluye la clave primaria en su tabla (es decir que el otro modelo con el que se relaciona utiliza la relación belongs to).	<pre>protected static \$_has_one = array(     'users' =&gt; array(         'model_to' =&gt;         'Auth\\Model_Auth_User',         'key_from' =&gt; 'id',         'key_to' =&gt; 'profile_id',         'cascade_save' =&gt; true,         'cascade_delete' =&gt; false     ), );</pre>
<b>Has Many</b>	Esta relación se utiliza en una relación <b>1-N</b> y no incluye la clave primaria en su tabla (es decir que el otro modelo con el que se relaciona utiliza la relación belongs to).	<pre>protected static \$_has_many = array(     'resources_downloaded'=&gt;array(         'model_to' =&gt;         'Model_Resources',         'key_from' =&gt; 'id',         'key_to' =&gt;         'download_user_id',         'cascade_delete' =&gt; false     ), );</pre>



<b>Many to Many</b>	Esta relación se utiliza en una relación N-N, ambos modelos deben contener esta relación.	<pre>protected static \$_many_many = array(     'projects' =&gt; array(         'key_from' =&gt; 'id',         'key_through_from' =&gt;         'auth_users_id',         'table_through' =&gt;         'users_projects',         'key_through_to' =&gt;         'project_id',         'model_to' =&gt;         'Model_Projects',         'key_to' =&gt; 'id',         'cascade_save' =&gt; true,         'cascade_delete' =&gt; false,     ), );</pre>
---------------------	---	--

**Tabla 4-1: Tabla Explicativa Modelos FuelPHP**

En la figura 4-1 se observa las distintas relaciones que puede tener FuelPHP con ejemplos de la aplicación desarrollada. Por ejemplo, se observa que un perfil tiene un usuario (has\_one) y el perfil pertenece a ese usuario (belongs to). También se observa que un usuario tiene muchos recursos descargados (has\_many). Por último, en la relación many many se observa como un usuario tiene muchos proyectos (esta relación también se encontraría en el modelo de proyecto, pero por cuestiones de espacio no se incluye en el ejemplo ya que el funcionamiento es el mismo).

### 4.1.3 Controladores y Enrutado

FuelPHP procesa todas las peticiones mediante el archivo *public/index.php* y luego estas peticiones son interpretadas por el archivo routes.php que se encuentra en *fuel/app/config*. FuelPHP localiza cada ruta de nuestra aplicación mediante una traducción dentro de este archivo, es decir FuelPHP capta la URL que hemos introducido y mira en el archivo config.php la traducción para localizar el controlador necesario encargado de la petición. Por ejemplo:

```
//url calendar
'calendar' => 'events/calendar/calendar',
'calendar/search' => 'events/calendar/search',
```

**Figura 4-3: Ejemplo traducción de ruta para el calendario de 4ResearchersSoft.**

FuelPHP trabaja con la directiva de *namespaces* por tanto en la traducción de la figura 4-3 trabajará de la siguiente forma:

- Buscará en la carpeta de los controladores.
- El controlador deberá incorporar el nombre del subdirectorío si estuviere en uno, en este caso, events.
- Buscará el archivo php de este controlador, en este caso, calendar.php el cual también deberá incorporarse en el nombre del controlador después del prefijo.
- Por ultimo buscará el método utilizado por la petición (GET,POST,etc) y tomara el prefijo que le sigue ,en este caso calendar.

```

class Controller_Events_Calendar extends Controller_Auth
{
    public function get_calendar()
    {
        $view = View::forge('events/event_index.twig');
        $user = Auth::get_user(); //we get users
        $view->projects = $user->projects;
        $view->type_events = Model_Typeevents::give_array_type($this->language_id);
        return Response::forge($view);
    }
}

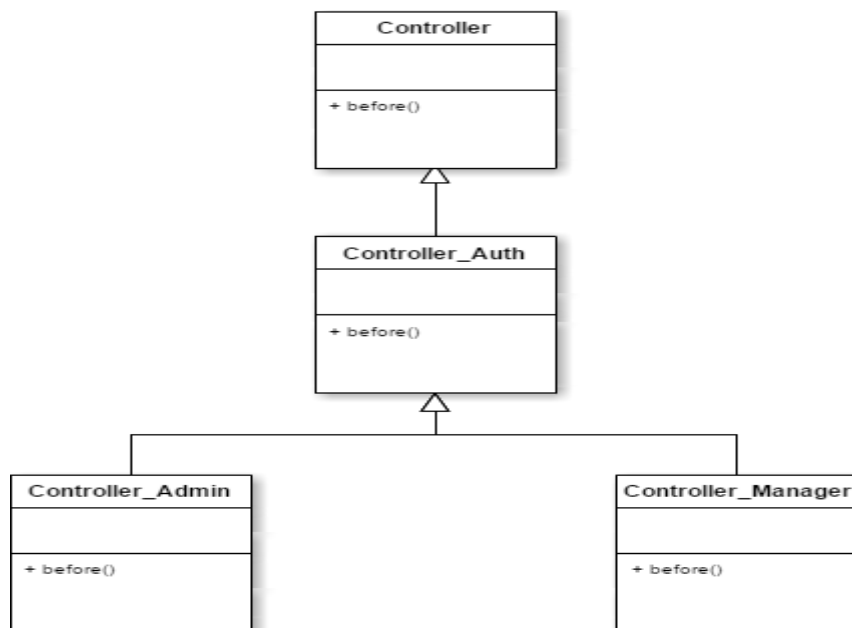
```

**Figura 4-4: Ejemplo controlador para el calendario de 4ResearchersSoft.**

De esta forma por ejemplo cuando se solicite <http://4researchersSoft.es/calendar> a través del navegador se está accediendo con el método GET a la ruta traducida y el controlador realizará las operaciones oportunas.

## 4.2 Desarrollo Subsistema Usuarios

En esta subsección se detalla el desarrollo del subsistema usuarios. En primer lugar, este subsistema tiene un punto crítico y es la diferenciación de los diferentes roles que hay en la aplicación para ello, se ha decidido extender del controlador por defecto de FuelPHP para permitir una clara diferenciación de los roles. Para ello se ha seguido la siguiente estructura:



**Figura 4-5: Diagrama clases roles 4ResearchersSoft.**

Como se puede observar en la Figura 4-5 se extiende del Controlador por defecto en FuelPHP creando un controlador de autenticación (Controller\_Auth), este controlador se encarga de saber mediante una extensión del método before de FuelPHP si el usuario está autenticado. Se deberá extender de este controlador para todas aquellas operaciones que realicen los miembros y los colaboradores. Por otro lado, el Controller\_Manager se encarga de verificar si el usuario autenticado es un responsable de proyecto por tanto se deberá extender de este cuando la operación la pueda realizar el responsable del proyecto.

Y por último el controlador Controller\_Admin es para las operaciones que realice el administrador.

Una vez creado esta estructura para los diferentes roles, se deberán crear los modelos, las vistas y los controladores asociados a los usuarios.

### 4.2.1 Ejemplo Modelo Subsistema Usuarios

A continuación, se muestra un ejemplo del Modelo para el perfil:

```
class Model_Auth_Profile extends Model_Soft
{
    /*Properties*/
    protected static $_table_name = 'auth_profile';
    protected static $_primary_key = array('id');

    //method calling method to array exclude this columns from database
    protected static $_to_array_exclude = array(
        'created_at',
        'updated_at'
    );

    protected static $_soft_delete = array(
        'deleted_field' => 'deleted_at',
        'mysql_timestamp' => true,
    );

    /*Relations*/
    protected static $_has_one = array(
        'users' => array(
            'model_to' => 'Auth\\Model_Auth_User',
            'key_from' => 'id',
            'key_to' => 'profile_id',
            'cascade_save' => true,
            'cascade_delete' => false
        ),
    );
};
```

Extendemos del Model Soft para permitir borrado logico

Nombre Tabla

Clave Primaria

Excluimos los campos que no nos interesen.

Campo deleted\_at de la base de datos para permitir borrado logico

Relacion 1-1

Figura 4-6: Ejemplo Modelo Perfil de Usuario

Como se muestra en la Figura 4-6 el modelo tiene varias particularidades en primer lugar, se extiende de Model\_Soft que es un tipo de modelo de FuelPHP que permite hacer **borrado lógico** es decir un borrado en el que se mantiene en la base de datos, pero no se muestra a los usuarios. Otra particularidad es el array `_to_array_exclude` que permite excluir columnas. Por último, se añade las relaciones que tenga el modelo en este caso la relación Has One.

## 4.2.2 Ejemplo Controlador Subsistema Usuarios

A continuación, se muestra un ejemplo del Controlador de usuarios que elimina a un usuario en la base de datos mediante AJAX:

```
public function delete_users()
{
    $response = array('id' => -1, 'message_body' => Lang::get('error_modal_body_default'));

    $id = Input::delete('id'); // Se recoge el Id del usuario desde el metodo Delete de HTTP

    //ve cannot remove admin so return false if that happen.
    if ($id < 1 || !$id || $id == 1) {
        return json_encode($response);
    } else {
        $user = \Auth\Model\Auth_User::find($id); // Se hace una busqueda a traves del ORM de FuelPHP con ese Id

        if (!$user || $user->is_permanent == 1) {
            return json_encode($response);
        }

        $user->delete(); // Se borra al usuario como el modelo extiende de Model_Soft se realiza un borrado logico
        $response['id'] = $id;
        $response['message body'] = Lang::get('delete_user_ok');
        return json_encode($response); // Se devuelve en formato JSON la respuesta del servidor
    }
}
```

Figura 4-7: Ejemplo Borrado Usuario desde el controlador

Como se muestra en la Figura 4-7 ese código del controlador se ejecutará cuando se realice una petición de tipo DELETE del usuario, recibiendo el identificador del usuario. Lo primero que se debe observar es como se obtiene el identificador a través de la clase Input de FuelPHP. Una vez obtenido el identificador del usuario se realiza una búsqueda con el método *find* de FuelPHP que permite encontrar un usuario por el identificador y por último se borra el usuario con el método delete (al ser un borrado lógico seguirá en la base de datos, pero no se mostrará) y se devuelve la respuesta en formato JSON para que el lado cliente la trate.

## 4.3 Desarrollo Subsistema Proyectos

En esta subsección se detalla el desarrollo del subsistema proyectos. Los proyectos es la entidad más importante de la aplicación ya que se relacionan con los recursos, usuarios y eventos. Los proyectos tienen recursos que se han de bloquear cuando estén siendo utilizados para ello se ha creado un manejador de archivos que se basa en el patrón de diseño *Singleton* [19] para restringir el acceso de estos recursos.

Singleton
- <u>singleton : Singleton</u>
- Singleton()
+ <u>getInstance() : Singleton</u>

Figura 4-8: Patrón de diseño Singleton

### 4.3.1 Ejemplo Manejador de Ficheros Subsistema Proyectos

A continuación, se muestra un ejemplo de la descarga de un archivo mediante el manejador de ficheros desarrollado:

```
public function download_file($resource)
{
    $message = array('status' => -1, 'message' => '');
    try {
        $message['status'] = 1;
        $resource->is_block = 1;
        $resource->user_download = \Auth\Auth::get_user();
        $resource->save();
        File::download($resource->path . $resource->metadata->title);
    } catch (InvalidPathException $e) {
        $message['message'] = Lang::get('download_server_err');
    } catch (InvalidArgumentException $e) {
        $message['message'] = Lang::get('download_server_err');
    } catch (FileAccessException $e) {
        $message['message'] = Lang::get('download_server_err');
    }
    return $message;
}
```

Accedemos al usuario logueado, asignamos ese recurso al usuario y actualizamos la informacion en la base de datos

Mediante la libreria que ofrece FuelPHP se descarga el recurso

Se tratan los diferentes errores que podrian suceder durante la descarga del recurso

Figura 4-9: Ejemplo descarga de un recurso

Como se muestra en la Figura 4-9 para la funcionalidad de descarga de un recurso, lo primero que se realiza es bloquear ese recurso para futuros usuarios, tras esto se asocia el usuario que está descargando el recurso al propio recurso para en un futuro él pueda subir el archivo y desbloquearlo, por último, se descarga el archivo con la ayuda del método *download* de FuelPHP.

## 4.4 Desarrollo Subsistema Estadísticas

En esta subsección se detalla el desarrollo del subsistema de estadísticas. En este subsistema se ha decidido utilizar como herramienta Google Charts [20] que permite realizar gráficos muy completos y visualmente atractivos para el usuario. El inconveniente de la utilización de esta herramienta es la necesidad de una conexión a internet para el acceso a los servidores de Google.

Esta herramienta permite a través de JavaScript dibujar estos gráficos y también personalizar los gráficos. También permite que los gráficos se adapten a las necesidades del desarrollador. Estos gráficos son interactivos gracias a que se procesan mediante HTML5 que proporciona una compatibilidad entre navegadores y una portabilidad de plataforma para dispositivos móviles. Los gráficos utilizan la clase DataTable que facilita el cambio de tipo de gráfico.

### 4.4.1 Ejemplo Grafico Tipo de Recurso

```
public function get_statstype()
{
    $response = array('id' => -1, 'message body' => Lang::get('error_modal_body_default'));
    $resources = Model_Resources::query()->get();
    $array_formats = array();

    if (!$resources) {
        return json_encode($response);
    }

    foreach ($resources as $resource) {
        $sin_array_format = false;

        foreach ($array_formats as $array_format) {
            if ($resource->metadata->format == $array_format['format']) {
                $sin_array_format = true;
                $array_format['rep'] = $array_format['rep'] + 1;
                break;
            }
        }

        if (!$sin_array_format) {
            $array_formats[] = array('format' => $resource->metadata->format, 'rep' => 1);
        }
    }

    $num_resources = count($resources);
    foreach ($array_formats as $array_format) {
        $array_format['rep'] = ($array_format['rep'] / $num_resources) * 100;
    }
    return json_encode($array_formats);
}
```

Consulta que devuelve todos los recursos que contiene el servidor.

Se recorre cada recurso y si existiese se aumenta en 1 las repeticiones de ese recurso

Si se cumple la condicion es la primera aparicion del tipo por tanto se añade

Se calcula el porcentaje total del tipo de recurso en el servidor.

Figura 4-10: Ejemplo Código Estadística Grafico Tipo de Recurso Lado Servidor

Como se muestra en la figura 4-10 para generar estadísticas en el lado del servidor el funcionamiento es igual que cualquier otro subsistema simplemente debemos realizar las tareas correspondientes a las estadísticas. En este caso lo primero que se realiza es una consulta que devuelve los recursos del servidor para calcular el porcentaje del tipo de archivos que hay en el servidor para la realización de un informe grafico de los recursos más utilizados en la aplicación. A continuación, se muestra el lado del cliente que utiliza la herramienta de Google Charts antes mencionada:

```
$.ajax({
    dataType: 'json',
    type: "GET",
    url: "/admin/statstype",
    success: function (data) {
        google.charts.load('current', {'packages':['corechart']});
        var array_google = [['Format', 'Reps']];
        data.forEach(function(element) {
            array_google.push([element['format'], element['rep']]);
        });
        google.charts.setOnLoadCallback(function() {
            drawChart(array_google, true);
        });
    }
});

function drawChart(array google, pie chart) {
    var data = google.visualization.arrayToDataTable(array_google);
    if (pie chart) {
        var chart = new google.visualization.PieChart(document.getElementById('piechart'));
        chart.draw(data);
    } else {
        var barchart_options = {
            'backgroundColor': 'transparent',
            'legend': 'none';
        };
        var barchart = new google.visualization.ColumnChart(document.getElementById('barchart'));
        barchart.draw(data, barchart_options);
    }
}
```

Se carga la API y el paquete corechart

Se establece un callback a la funcion para dibujar el grafico que se desee

Se pasa el array con los datos del grafico al tipo de dato de Google DataTable

Se dibuja el tipo de grafico deseado en este caso un grafico de tipo tarta

Figura 4-11: Ejemplo Código Estadística Grafico Tipo de Recurso Lado Cliente

En la figura 4-11 se puede observar cómo utilizar esta herramienta para dibujar gráficos.

## 5 Integración, pruebas y resultados

En esta sección se detallan las pruebas realizadas a la aplicación desarrollada. Se realizaron varias pruebas para comprobar el funcionamiento de la aplicación, estas pruebas se dividen en tres categorías: Pruebas Unitarias, Pruebas Funcionales y Pruebas de *carga/estrés*.

### 5.1 Pruebas Unitarias

Para cada módulo de la aplicación se ha comprobado el comportamiento frente al paso de argumentos tanto validos como inválidos y se han analizado los valores devueltos. También se comprueban las diferentes rutas de la aplicación y el tratamiento de los datos que el cliente envía al servidor.

### 5.2 Pruebas Funcionales

En esta sección se muestran los resultados de las pruebas funcionales es decir aquellas pruebas que aseguran el cumplimiento de los requisitos funcionales de la aplicación. Para ello se muestra una tabla por cada requisito funcional que contiene el identificador del requisito funcional a probar, una breve descripción de la prueba, los resultados esperados de la prueba y si ha superado la prueba. A continuación, se muestran las pruebas funcionales más importantes realizadas (Para ver todas, ir al **Anexo B**):

Identificador Requisito Funcional	RF2
Descripción	Esta prueba se encarga de comprobar que el registro por parte del administrador funciona correctamente.
Entrada	Los datos del usuario, como el email, la contraseña etc.
Resultado Esperado	El usuario es registrado en la aplicación con el rol deseado y se ha insertado en la base de datos.
Prueba Realizada con éxito	Si

Tabla 5-1: Tabla Prueba Funcional Requisito Funcional 2

Identificador Requisito Funcional	RF3
Descripción	Esta prueba se encarga de comprobar que el login por parte de un usuario funciona correctamente
Entrada	Email y contraseña del usuario.
Resultado Esperado	El usuario se loguea correctamente y se crea una variable de sesión en el navegador.
Prueba Realizada con éxito	Si

Tabla 5-2: Tabla Prueba Funcional Requisito Funcional 3

Identificador Requisito Funcional	RF10
Descripción	Esta prueba se encarga de comprobar que la búsqueda de usuarios se realiza de una forma correcta y eficaz.
Entrada	Los datos por los que se quiera filtrar.
Resultado Esperado	La aplicación muestra un listado de usuarios filtrados correctamente por los datos de filtrado.
Prueba Realizada con éxito	Si

**Tabla 5-3: Tabla Prueba Funcional Requisito Funcional 10**

Identificador Requisito Funcional	RF11
Descripción	Esta prueba se encarga de comprobar que se dan de alta a nuevos proyectos.
Entrada	Los datos del proyecto, como su nombre o miembros.
Resultado Esperado	El proyecto es creado con toda la información y miembros en la base de datos.
Prueba Realizada con éxito	Si

**Tabla 5-4: Tabla Prueba Funcional Requisito Funcional 11**

Identificador Requisito Funcional	RF12
Descripción	Esta prueba se encarga de comprobar que los proyectos se cierran correctamente.
Entrada	Identificador del proyecto a cerrar.
Resultado Esperado	El proyecto se cierra correctamente y por tanto no se puede trabajar con él hasta que se abra de nuevo.
Prueba Realizada con éxito	Si

**Tabla 5-5: Tabla Prueba Funcional Requisito Funcional 12**



Identificador Requisito Funcional	RF15
Descripción	Esta prueba se encarga de comprobar que el administrador puede dar de alta a un usuario en un proyecto.
Entrada	Email del usuario a añadir.
Resultado Esperado	El usuario se añade al proyecto correctamente.
Prueba Realizada con éxito	Si

**Tabla 5-6: Tabla Prueba Funcional Requisito Funcional 15**

Identificador Requisito Funcional	RF19
Descripción	Esta prueba se encarga de comprobar que el administrador da de alta un evento en un proyecto con los usuarios que tienen que asistir al evento.
Entrada	Los datos del evento, el proyecto asociado y sus miembros.
Resultado Esperado	El evento es dado de alta, se asocia tanto los usuarios que tienen que asistir como el proyecto seleccionado.
Prueba Realizada con éxito	Si

**Tabla 5-7: Tabla Prueba Funcional Requisito Funcional 19**

Identificador Requisito Funcional	RF24
Descripción	Esta prueba se encarga de comprobar que el material del proyecto se sube correctamente al servidor.
Entrada	El recurso que se quiera almacenar.
Resultado Esperado	El recurso se almacena correctamente cumpliendo las restricciones (100Mb máximo y no se permite la extensión .exe)
Prueba Realizada con éxito	Si

**Tabla 5-8: Tabla Prueba Funcional Requisito Funcional 24**

Identificador Requisito Funcional	RF26
Descripción	Esta prueba se encarga de comprobar que el material del proyecto se descarga correctamente.
Entrada	Identificador del recurso a descargar.
Resultado Esperado	El recurso se descarga correctamente en el cliente y se bloquea para futuros accesos para aquellos que no estén trabajando con él.
Prueba Realizada con éxito	Si

**Tabla 5-9: Tabla Prueba Funcional Requisito Funcional 26**

Identificador Requisito Funcional	RF27
Descripción	Esta prueba se encarga de comprobar que los metadatos de los recursos se añaden automáticamente correctamente.
Entrada	Los metadatos del recurso.
Resultado Esperado	Los metadatos se guardan correctamente en la base de datos.
Prueba Realizada con éxito	Si

**Tabla 5-10: Tabla Prueba Funcional Requisito Funcional 27**

Identificador Requisito Funcional	RF31
Descripción	Esta prueba se encarga de comprobar que el administrador puede visualizar correctamente el grafico del número de recursos activos por mes.
Entrada	Datos estadísticos del material.
Resultado Esperado	El administrador puede visualizar correctamente el grafico con los recursos activos
Prueba Realizada con éxito	Si

**Tabla 5-11: Tabla Prueba Funcional Requisito Funcional 31**

## 5.3 Pruebas Carga/Estrés

En esta sección se muestran los resultados de las pruebas de **carga y estrés** realizadas a la aplicación desarrollada. Las pruebas han sido realizadas mediante la herramienta *Apache JMeter* [21]. La finalidad es mostrar el comportamiento del servidor de la aplicación ante una carga de trabajo notable. Se ha realizado una fase donde los usuarios se autentifiquen y ver cómo responde la aplicación y otra fase donde se prueban las tareas del administrador como puede ser la creación de usuarios. A continuación, se muestra los resultados obtenidos en la primera fase:

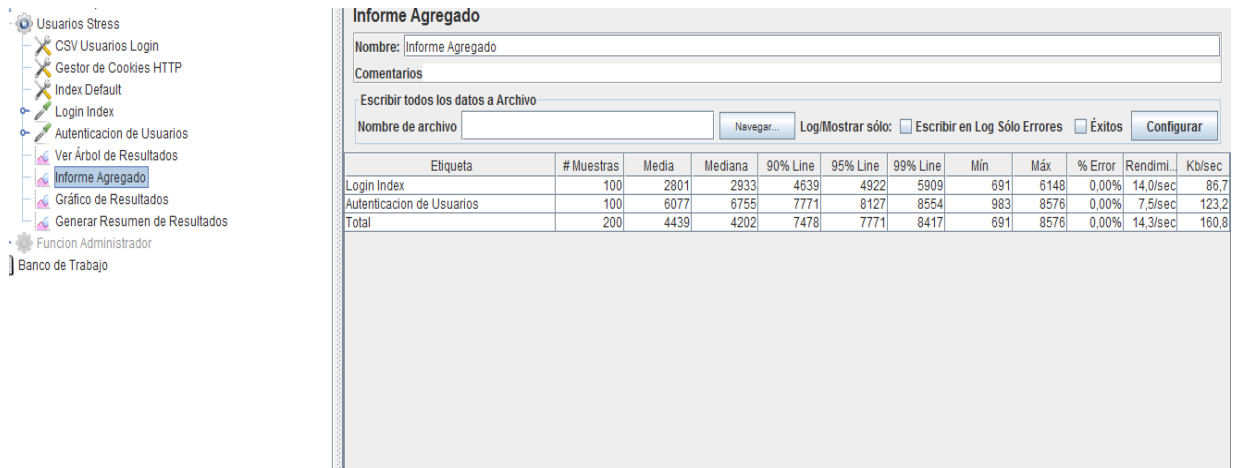


Figura 5-1: Resultados JMeter fase autenticación

Como se puede observar en la figura 5-12 se ha realizado una prueba para 100 usuarios, el rendimiento para este número de usuarios es asumible, pero si se aumenta drásticamente los usuarios, el sistema empieza a colapsar debido a que las pruebas se han realizado en un ordenador personal no muy potente. Por último, se muestra la fase de las tareas de administración:

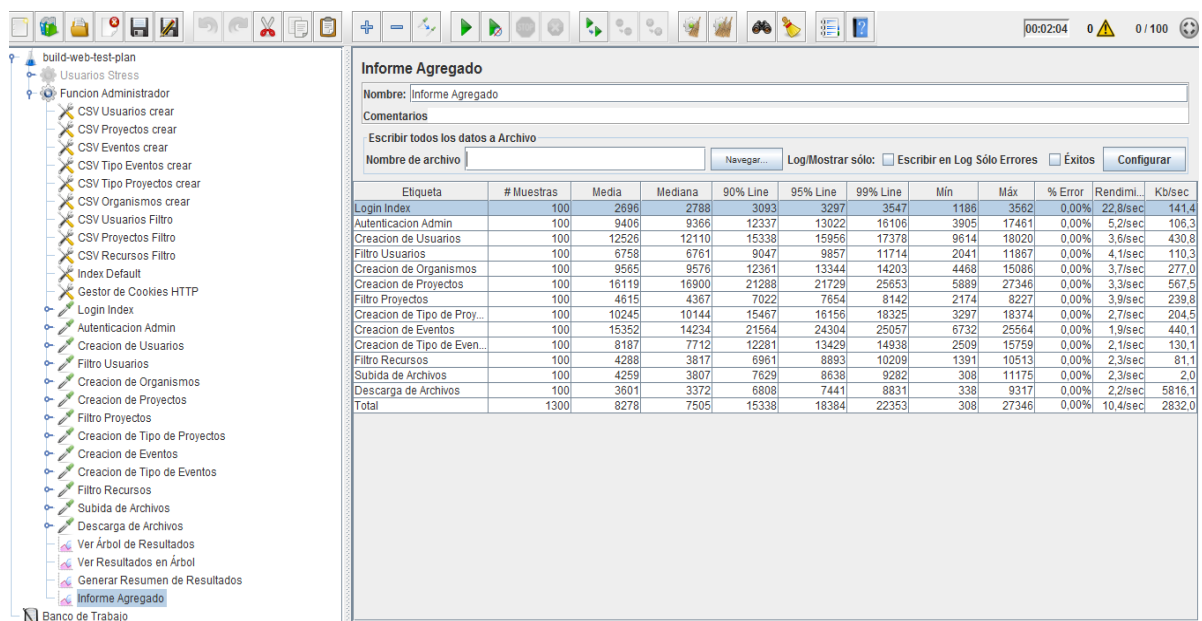


Figura 5-2: Resultados JMeter fase administración



## 6 Conclusiones y trabajo futuro

---

### 6.1 Conclusiones

En este trabajo se ha desarrollado una aplicación web pasando por todas las fases del ciclo de vida de un proyecto. Los resultados obtenidos tras el desarrollo de este proyecto han sido satisfactorios puesto que se ha realizado una aplicación web que cumple con el objetivo (facilitar la gestión de un grupo de investigación.)

La aplicación cumple todos los requisitos tanto funcionales como no funcionales descritos en el apartado de diseño incluyendo entre otros el registro de eventos, proyectos, usuarios y permitiendo ver estadísticas generadas por la aplicación.

Por último, centrándonos en el aprendizaje obtenido, podemos destacar el uso de FuelPHP, Twig, Bootstrap, AJAX, HTML5 Y PHP herramientas que facilitan el desarrollo de la aplicación web.

### 6.2 Trabajo futuro

La primera versión del proyecto es una versión totalmente operativa y aunque tiene un diseño completamente *responsive* se trabajará más en el diseño para poder atraer a más usuarios.

Por otro lado, la privacidad en los datos es muy importante hoy en día y aunque la aplicación posee seguridad para ciertos ataques como el *CSRF* se trabajará en la mejora del trato de los datos que sean sensibles añadiendo nuevas medidas de seguridad, por ejemplo, protección contra *ataques de fuerza bruta* para que un usuario maligno capture la contraseña de un usuario legítimo.

Por último, se trabajará en una mejora del tratamiento de los recursos añadiendo la posibilidad de recuperar versiones anteriores para así asemejar el trabajo de los recursos a un control de versiones, ya que se podría trazar perfectamente los cambios realizados por cada investigador.



# Referencias

---

- [1] Internet ya [Última Visita 18/09/2016]: <http://www.internetya.co/ventajas-y-beneficios-de-las-aplicaciones-web/>
- [2] ActiveCollab. [Última Visita 20/09/2016]: <https://www.activecollab.com/>
- [3] BaseCamp. [Última Visita 24/09/2016]: <https://basecamp.com/>
- [4] AceProject [Última Visita 25/09/2016]: <http://www.aceproject.com/>
- [5] Twig [Última Visita 26/9/2016]: [https://en.wikipedia.org/wiki/Twig\\_\(template\\_engine\)](https://en.wikipedia.org/wiki/Twig_(template_engine))
- [6] Bootstrap [Última visita 26/09/2016]: [https://es.wikipedia.org/wiki/Twitter\\_Bootstrap](https://es.wikipedia.org/wiki/Twitter_Bootstrap)
- [7] Luc Van Lancker. JQuery el framework JavaScript de la web 2.0. 1º Ed. 2012.
- [8] AJAX [Última visita 27/09/2016]: <http://www.w3schools.com/ajax/>
- [9] MVC [Última visita 27/09/2016]:  
<https://es.wikipedia.org/wiki/Modelo%E2%80%93vista%E2%80%93controlador>
- [10] Ross Teedie. Learning FuelPHP for Effective PHP Development. 2013
- [11] HMVC [Última visita 27/09/2016]:  
[https://en.wikipedia.org/wiki/Hierarchical\\_model%E2%80%93view%E2%80%93controller](https://en.wikipedia.org/wiki/Hierarchical_model%E2%80%93view%E2%80%93controller)
- [12] Patrón Active Record [Última visita 29/09/2016]:  
[https://es.wikipedia.org/wiki/Active\\_record](https://es.wikipedia.org/wiki/Active_record)
- [13] Modelo en cascada [Última visita 29/09/2016]:  
<http://ele-mariamoliner.dyndns.org/~fperal/proy/ingenieriaSW.pdf>
- [14] Paul Kimmel. Manual de UML. Capítulo 2. 2006
- [15] Modelo Entidad-Relacion [Última visita 29/09/2016]:  
<http://www.ual.es/~mtorres/BD/bdtransp2.pdf>
- [16] Técnicas multilenguaje base de datos [Última visita 10/10/2016]:  
<http://www.apphp.com/tutorials/index.php?page=multilanguage-database-design-in-mysql>
- [17] IDE [Última visita 14/10/2016]:  
[https://es.wikipedia.org/wiki/Entorno\\_de\\_desarrollo\\_integrado](https://es.wikipedia.org/wiki/Entorno_de_desarrollo_integrado)
- [18] JetBrains PhpStorm [Última visita 14/10/2016]:  
<https://www.jetbrains.com/phpstorm/?fromMenu>
- [19] Laurent Debrauwer, Yannick Evain. Patrones de diseño en PHP: Los 23 modelos de diseño: descripciones y soluciones ilustradas en UML2 y PHP Capitulo Patrón Singleton
- [20] Google Charts [Última visita 12/11/2016]:  
<https://developers.google.com/chart/interactive/docs/>
- [21] Bayo Erinle. Performance Testing with JMeter 2º Ed. 2015.
- [22] XAMPP [Última visita 02/12/2016]: <https://www.apachefriends.org/es/index.html>
- [23] Navicat [Última visita 02/10/2016]: <https://www.navicat.com/es>

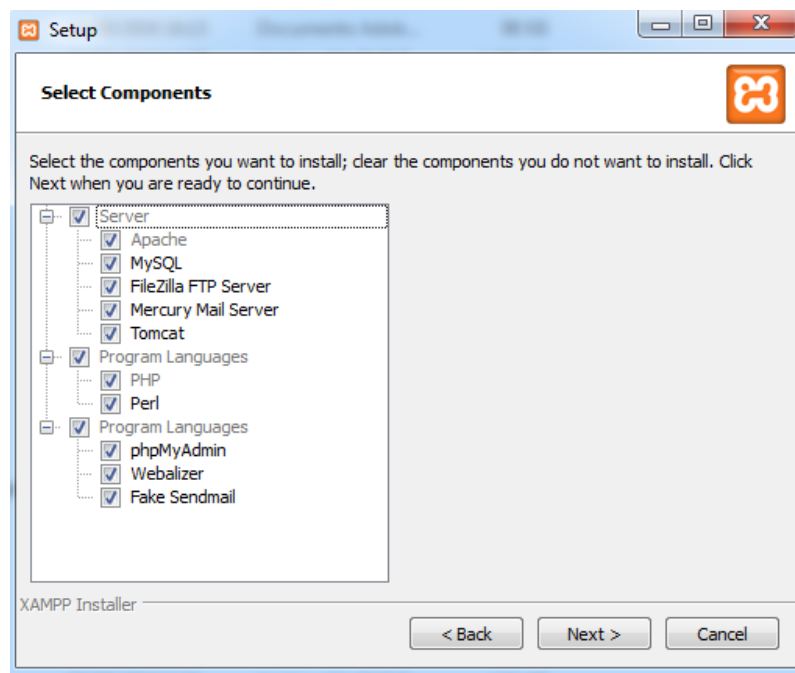




## Anexos

### A Manual de instalación

Para poder instalar correctamente la aplicación 4ResearchersSoft se necesitará descargar XAMPP desde <https://www.apachefriends.org/es/download.html>. XAMPP es un servidor independiente de plataforma que incluye Apache, PHP y MySQL [22]. Por tanto, en la instalación se deberá tener marcadas estas tres opciones (Ver Figura A-1). Como FuelPHP requiere una versión sucesiva a la 5.3, la versión de XAMPP instalada deberá cumplir este requisito, la versión seleccionada para las pruebas de la aplicación ha sido la 5.5.37 y el sistema operativo elegido Windows 7. La localización de XAMPP deberá tener todos los permisos para que no haya ningún problema con ellos, en el caso de ejemplo se instalará en la ruta: C:\xampp. Para comprobar que se ha instalado correctamente y se dispone de una versión superior a la 5.3 podemos ejecutar el comando `php -v` desde la consola de Windows.



**Figura A-1: Panel Instalación XAMPP**

Tras la instalación de XAMPP se deberá configurar apache y añadir al servidor un host virtual en local para que la aplicación funcione correctamente.

- Para añadir un host virtual en local se deberá localizar el archivo hosts en la ruta donde se disponga el sistema operativo Windows. En el caso de ejemplo de este manual el archivo se encuentra en: **C:\Windows\System32\drivers\etc**
- En el archivo hosts se deberá añadir las siguientes líneas:  
127.0.0.1      tfg  
127.0.0.1      localhost

- A continuación, se deberá dirigir al servidor apache y localizar el archivo httpd-vhost.conf en el caso del ejemplo de este manual el archivo se encuentra en la ruta **C:\xampp\apache\conf\extra** y se deberá añadir las siguientes líneas:

```
<VirtualHost *:80>
    DocumentRoot "C:/xampp/htdocs/4ResearchersSoft/public"
    ServerName tfg
</VirtualHost>
```

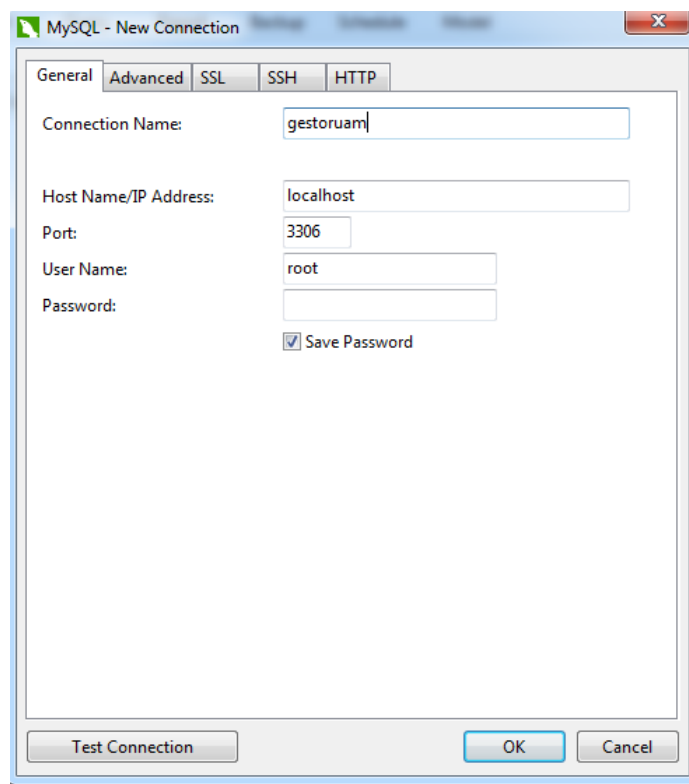
Estas líneas indican el puerto virtual en este caso el puerto 80 (HTTP), la carpeta publica a la que tendrá acceso el navegador y el nombre del servidor. Tras realizar esto se deberá reiniciar apache desde la consola de administración de XAMPP. Si se quiere habilitar algunas extensiones como OpenSSL se deberá dirigir a la carpeta que contenga php, en el caso de ejemplo es: C:\xampp\php. Una vez localizada la carpeta se tendrá que abrir el archivo php.ini y descomentar la línea **extension=php\_openssl.dll**.

También en php.ini se deberá configurar las variables **upload\_max\_filesize**, **post\_max\_size**, **max\_file\_uploads**, estas variables se podrán configurar como se desee dependiendo del servidor en el que se aloje la aplicación, en este caso se les ha dado el valor de 300MB, 100MB y 3.

En este punto de la instalación ya tenemos nuestro servidor configurado, pero aún falta configurar la base de datos y FuelPHP.

Para configurar la base de datos se ha utilizado Navicat como gestor de base de datos [23]. Navicat soporta conexiones SSH y bases de datos MySQL. Para instalar Navicat se deberá acceder a su página web oficial y descargar desde <https://www.navicat.com/es/download>.

Tras instalar Navicat debemos crear una base de datos con una conexión MySQL como en la figura A-2:



**Figura A-2: Creación base de datos MySQL con Navicat**

Como se observa en la Figura A-2 en el ejemplo de este manual se ha decidido llamar a la conexión gestoruaam, escuchará en el puerto 3306 ya que es MySQL y tendrá como usuario root y no dispondrá de contraseña.

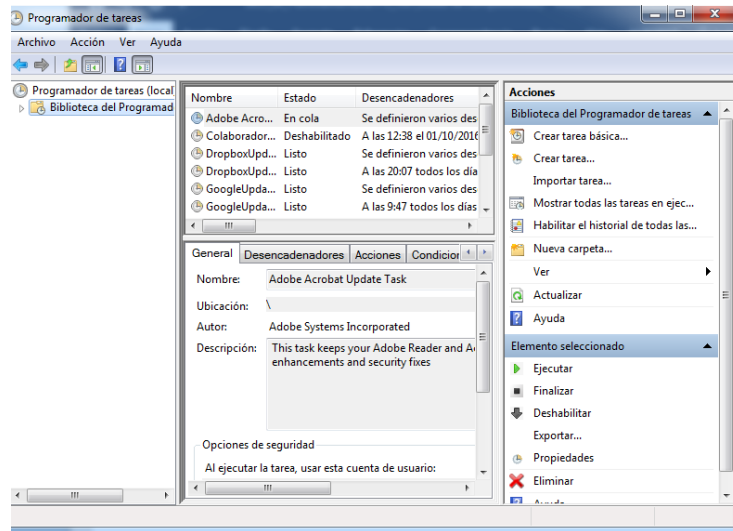
Tras crear la base de datos, se debe configurar el framework FuelPHP. Los pasos a seguir para configurar el framework son los siguientes:

- En la carpeta raíz del proyecto añadir en el fichero **composer.json** la siguiente línea donde se encuentra require: "twig/twig": "1.14.1" para que Twig sea obligatorio en nuestro proyecto.
- Tras seguir el paso anterior, se deberá ejecutar el comando desde la consola de Windows: *php composer.phar update* para instalar Twig.
- Por último, se deberá localizar el archivo config.php en la carpeta app/config y añadir los siguientes cambios en el array:
  - 'cache\_lifetime' => 3600
  - 'language' => 'en'
  - 'language\_fallback' => 'en'
  - 'csrf\_auto\_token' => true,
  - 'csrf\_token\_key' => 'fuel\_csrf\_token',
  - 'csrf\_expiration' => 1800,
  - 'always\_load' => array (
    - 'packages' => array (
      - 'orm',
      - 'parser',
      - 'email',
  - ),
  - 'modules' => array(
    - 'auth',
    - 'validation',
  - ),

Estos cambios en el archivo de configuración se realizan para habilitar opciones necesarias para el funcionamiento de la aplicación y habilitar paquetes como el ORM y el email.

El último paso que se deberá seguir es la configuración de una tarea programada para la ejecución de un **script** que se encargará de la desactivación de los colaboradores externos para ello se deberá:

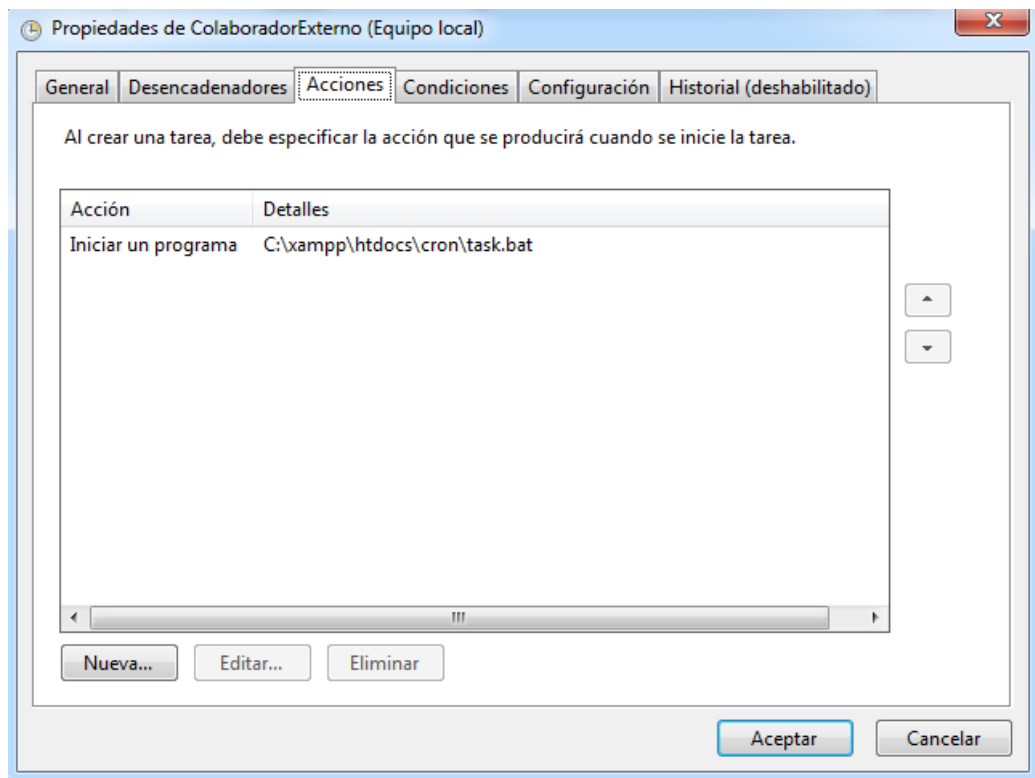
- Acceder al Panel de control de Windows 7 y desde herramientas administrativas dirigirse al Programador de Tareas como se puede ver en la Figura (A-3).



**Figura A-3: Programador de Tareas Windows 7**

- Crear una carpeta llamada **cron** en **C:\xampp\htdocs** y crear un script que realice las siguientes operaciones:  

```
cd \xampp\htdocs\4ResearchersSoft
php oil refine active
```
- Tras esto por último vincularemos el script a la tarea a la hora que se desee (Ver Figura A-4) y la tarea estará lista para su ejecución.



**Figura A-4: Script Vinculado por el programador de tareas de Windows 7**

## B Anexo Pruebas Funcionales

B. Identificador Requisito Funcional	RF4
Descripción	Esta prueba se encarga de comprobar que se cambia con éxito la contraseña tanto desde el panel de administración como de la opción recuperar contraseña.
Entrada	Si se cambia desde el panel de administración, la entrada será la contraseña nueva. Si se cambia con la opción recuperar contraseña la entrada será el email del usuario que se quiera cambiar de contraseña.
Resultado Esperado	La contraseña se cambia con éxito, y en caso de que haya sido solicitado el cambio mediante la opción de recuperar email con la nueva contraseña del usuario.
Prueba Realizada con éxito	Si

Tabla B-1: Tabla Prueba Funcional Requisito Funcional 4

Identificador Requisito Funcional	RF5
Descripción	Esta prueba se encarga de comprobar que se puede actualizar el perfil de un usuario correctamente.
Entrada	Los datos del usuario a cambiar.
Resultado Esperado	El perfil del usuario se ha cambiado correctamente en la base de datos.
Prueba Realizada con éxito	Si

Tabla B-2: Tabla Prueba Funcional Requisito Funcional 5

Identificador Requisito Funcional	RF6
Descripción	Esta prueba se encarga de comprobar que se puede dar de baja a un usuario de la aplicación.
Entrada	El Identificador del usuario a ser eliminado.
Resultado Esperado	El usuario es eliminado con éxito.
Prueba Realizada con éxito	Si

Tabla B-3: Tabla Prueba Funcional Requisito Funcional 6

Identificador Requisito Funcional	RF7
Descripción	Esta prueba se encarga de comprobar que el administrador puede editar tanto la información como el rol de un usuario.
Entrada	Los datos del usuario, como el nombre o el rol de usuario.
Resultado Esperado	El usuario es editado con éxito en la base de datos.
Prueba Realizada con éxito	Si

**Tabla B-4: Tabla Prueba Funcional Requisito Funcional 7**

Identificador Requisito Funcional	RF8
Descripción	Esta prueba se encarga de comprobar que el administrador puede dar de alta nuevos organismos.
Entrada	El nombre del organismo.
Resultado Esperado	El organismo se ha dado de alta correctamente en la base de datos.
Prueba Realizada con éxito	Si

**Tabla B-5: Tabla Prueba Funcional Requisito Funcional 8**

Identificador Requisito Funcional	RF9
Descripción	Esta prueba se encarga de comprobar que el administrador puede dar de baja a un organismo.
Entrada	El identificador del organismo.
Resultado Esperado	El organismo es eliminado, a menos, que ese organismo pertenezca a un colaborador externo aun en activo.
Prueba Realizada con éxito	Si

**Tabla B-6: Tabla Prueba Funcional Requisito Funcional 9**

Identificador Requisito Funcional	RF13
Descripción	Esta prueba se encarga de comprobar que el administrador puede dar de alta nuevos tipos de proyectos.
Entrada	El nombre del tipo de proyecto.
Resultado Esperado	El tipo de proyecto se ha dado de alta correctamente en la base de datos.
Prueba Realizada con éxito	Si

**Tabla B-7: Tabla Prueba Funcional Requisito Funcional 13**

Identificador Requisito Funcional	RF14
Descripción	Esta prueba se encarga de comprobar que el administrador puede dar de baja a un tipo de proyecto
Entrada	El identificador del tipo de proyecto.
Resultado Esperado	El tipo de proyecto es eliminado, a menos, que ese tipo de proyecto tenga un proyecto asociado.
Prueba Realizada con éxito	Si

**Tabla B-8: Tabla Prueba Funcional Requisito Funcional 14**

Identificador Requisito Funcional	RF16
Descripción	Esta prueba se encarga de comprobar que el administrador puede dar de baja a un usuario en un proyecto.
Entrada	Identificador del usuario a dar de baja.
Resultado Esperado	El usuario es dado de baja del proyecto.
Prueba Realizada con éxito	Si

**Tabla B-9: Tabla Prueba Funcional Requisito Funcional 16**

Identificador Requisito Funcional	RF17
Descripción	Esta prueba se encarga de comprobar que el material del proyecto se sube a la localización correcta.
Entrada	Recurso a añadir.
Resultado Esperado	El recurso se encuentra localizado en la ubicación deseada.
Prueba Realizada con éxito	Si

**Tabla B-10: Tabla Prueba Funcional Requisito Funcional 17**

Identificador Requisito Funcional	RF18
Descripción	Esta prueba se encarga de comprobar que la información del proyecto se edita correctamente.
Entrada	Los datos del proyecto a cambiar
Resultado Esperado	El proyecto se edita correctamente con los nuevos datos.
Prueba Realizada con éxito	Si

**Tabla B-11: Tabla Prueba Funcional Requisito Funcional 18**

Identificador Requisito Funcional	RF20
Descripción	Esta prueba se encarga de comprobar que el administrador da de baja un evento en un proyecto.
Entrada	Identificador del evento a dar de baja.
Resultado Esperado	El evento es dado de baja junto a todos sus usuarios.
Prueba Realizada con éxito	Si

**Tabla B-12: Tabla Prueba Funcional Requisito Funcional 20**

Identificador Requisito Funcional	RF21
Descripción	Esta prueba se encarga de comprobar que el administrador edita correctamente la información de un evento.
Entrada	Los datos del evento a modificar, por ejemplo, los asistentes al evento.
Resultado Esperado	El evento se edita correctamente con su nueva información.
Prueba Realizada con éxito	Si

**Tabla B-13: Tabla Prueba Funcional Requisito Funcional 21**

Identificador Requisito Funcional	RF22
Descripción	Esta prueba se encarga de comprobar que el administrador puede dar de alta nuevos tipos de eventos.
Entrada	El nombre del tipo de eventos.
Resultado Esperado	El tipo de evento se ha dado de alta correctamente en la base de datos.
Prueba Realizada con éxito	Si

**Tabla B-14: Tabla Prueba Funcional Requisito Funcional 22**



Identificador Requisito Funcional	RF23
Descripción	Esta prueba se encarga de comprobar que el administrador puede dar de baja a un tipo de evento.
Entrada	El identificador del tipo de evento.
Resultado Esperado	El tipo de evento es eliminado, a menos, que ese tipo de evento tenga un evento asociado.
Prueba Realizada con éxito	Si

**Tabla B-15: Tabla Prueba Funcional Requisito Funcional 23**

Identificador Requisito Funcional	RF25
Descripción	Esta prueba se encarga de comprobar que el administrador elimine el recurso que desee
Entrada	Identificador del recurso a eliminar.
Resultado Esperado	El recurso es eliminado por completo tanto de la base de datos como de su ubicación.
Prueba Realizada con éxito	Si

**Tabla B-16: Tabla Prueba Funcional Requisito Funcional 25**

Identificador Requisito Funcional	RF28
Descripción	Esta prueba se encarga de comprobar que la búsqueda de proyectos se realiza de una forma correcta y eficaz.
Entrada	Los datos por los que se quiera filtrar.
Resultado Esperado	La aplicación muestra un listado de proyectos filtrados correctamente por los datos de filtrado.
Prueba Realizada con éxito	Si

**Tabla B-17: Tabla Prueba Funcional Requisito Funcional 28**

Identificador Requisito Funcional	RF29
Descripción	Esta prueba se encarga de comprobar que la búsqueda de eventos se realiza de una forma correcta y eficaz.
Entrada	Los datos por los que se quiera filtrar.
Resultado Esperado	La aplicación muestra un listado de eventos filtrados correctamente por los datos de filtrado.
Prueba Realizada con éxito	Si

**Tabla B-18: Tabla Prueba Funcional Requisito Funcional 29**

<b>Identificador Requisito Funcional</b>	<b>RF30</b>
<b>Descripción</b>	Esta prueba se encarga de comprobar que la búsqueda de material se realiza de una forma correcta y eficaz.
<b>Entrada</b>	Los datos por los que se quiera filtrar.
<b>Resultado Esperado</b>	La aplicación muestra un listado de recursos filtrados correctamente por los datos de filtrado.
<b>Prueba Realizada con éxito</b>	Si

**Tabla B-19: Tabla Prueba Funcional Requisito Funcional 30**

<b>Identificador Requisito Funcional</b>	<b>RF32</b>
<b>Descripción</b>	Esta prueba se encarga de comprobar que el administrador puede visualizar correctamente el grafico de estadísticas del material por tipo.
<b>Entrada</b>	Datos estadísticos del material.
<b>Resultado Esperado</b>	El administrador puede visualizar correctamente el grafico con el porcentaje correctamente.
<b>Prueba Realizada con éxito</b>	Si

**Tabla B-20: Tabla Prueba Funcional Requisito Funcional 32**